



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques

Universitat de Barcelona

**DISEÑO DE UN INVENTARIO WEB DE
PRODUCTOS QUÍMICOS PARA LA FACULTAD
DE QUÍMICA**

Javier Sánchez Velamazán

Director: Simone Balocco

Realizado en: Departament de
Matemàtica Aplicada i Anàlisi. UB

Barcelona, 22 de Mayo de 2015

Índice

| | |
|--|----|
| | 5 |
| Abstract (Castellano)..... | 6 |
| Abstract (Català)..... | 7 |
| Abstract (English)..... | 8 |
| 1.Introducción..... | 9 |
| 1.1.Inventario de productos químicos..... | 9 |
| 1.1.1.Sustancias Químicas..... | 9 |
| 1.1.2.Número de registro CAS..... | 10 |
| 1.1.3.Seguridad..... | 12 |
| 1.2.Motivación..... | 14 |
| 1.3.Objetivos..... | 14 |
| 1.4.Distribución de la Memoria..... | 16 |
| 2.Análisis de requerimientos..... | 17 |
| 2.1.Análisis de requerimientos y funcionalidades..... | 17 |
| 2.2.Requisitos funcionales..... | 17 |
| 2.2.1.Casos de uso para el usuario Anónimo..... | 18 |
| 2.2.2.Casos de uso para el usuario Profesor..... | 18 |
| 2.2.3.Casos de uso para el usuario Técnico de Laboratorio..... | 19 |
| 2.2.4.Casos de uso para el usuario Administrador..... | 21 |
| 2.3.Requisitos no funcionales..... | 22 |
| 3.Diseño del modelo de datos..... | 23 |
| 3.1.Jerarquía de productos en la aplicación..... | 23 |
| 3.2.Modelo Entidad / Relación (ER)..... | 24 |
| 3.3.Modelo de Datos..... | 26 |
| 4.Tecnología utilizada..... | 27 |
| 4.1.Servidor LAMP - Linux / Apache / MySQL / PHP..... | 27 |
| 4.2.Composer..... | 28 |
| 4.3.Framework PHP - Laravel..... | 28 |
| 4.4.Framework HTML & CSS - Bootstrap..... | 29 |
| 4.5.Repositorio de código - Github..... | 31 |
| 4.6.Metodologías Ágiles - Scrum y Kanban..... | 31 |
| 5.Laravel a fondo..... | 33 |

| | |
|--|----|
| 5.1.Laravel - The framework for web artisans..... | 33 |
| 5.2.Frameworks PHP..... | 33 |
| 5.3.¿Porque Laravel?..... | 38 |
| 5.4.Patrones del desarrollo web y buenas practicas en Laravel..... | 39 |
| 5.4.1. “Single responsibility patern” | 39 |
| 5.4.2. “Convention over configuration” | 40 |
| 5.4.3. “Don't repeat yourself” | 41 |
| 5.5.Composer & Laravel..... | 41 |
| 5.6.Artisan..... | 43 |
| 5.7.Migrations..... | 43 |
| 5.7.1.¿Que son las migraciones?..... | 43 |
| 5.7.2.Crear y ejecutar migraciones..... | 45 |
| 5.7.3.Seeds..... | 47 |
| 5.8.Eloquent..... | 48 |
| 5.9.RESTful routing..... | 48 |
| 5.10.Blade Templates..... | 49 |
| 5.11.Seguridad..... | 50 |
| 5.11.1.Protección de rutas y autenticación de usuarios..... | 50 |
| 5.11.2.Seguridad de la aplicación..... | 51 |
| 5.12.Laravel - El framework PHP mas famoso..... | 51 |
| 6.Arquitectura..... | 53 |
| 6.1.Modelos..... | 53 |
| 6.2.Vistas..... | 53 |
| 6.3.Controladores..... | 54 |
| 7.Planificación..... | 55 |
| 8.Tests realizados..... | 57 |
| 9.Conclusiones..... | 60 |
| 9.1.Mejoras propuestas y trabajo futuro..... | 60 |
| 9.1.1.Mejoras en el Cacheado del sistema..... | 60 |
| 9.1.2.Mejora en el Registro de logs..... | 61 |
| 9.1.3.Mejoras en los buscadores..... | 61 |
| 9.1.4.Creación y automatización de informes..... | 61 |
| 9.1.5.Ampliación a diferentes laboratorios..... | 61 |
| 9.1.6.Sistema de préstamos..... | 62 |

| | |
|--------------------------------------|----|
| 10. Bibliografía..... | 63 |
| 11. Anexo..... | 67 |
| 11.1. Proxy inverso servidor UB..... | 67 |
| 11.2. Configuración Apache..... | 68 |
| 11.3. Manual de usuario..... | 69 |
| 11.4. Casos de uso textuales..... | 79 |

Abstract (Castellano)

El proyecto documentado que se desarrollará en este Trabajo Fin de Grado consiste en un sistema de gestión mediante una aplicación web del inventario de productos químicos de la Facultad de Química. El proyecto consiste en el diseño e implementación de una aplicación web que se adapte a las necesidades de la Facultad de Química y permita gestionar diversos inventarios de productos con varios niveles de jerarquía de producto, diferentes niveles de acceso, modificación y consulta mediante la plataforma web.

Mediante esta herramienta, se podrá mantener un control y gestionar los productos que se encuentran en los diferentes almacenes, sus características y relaciones, así como la administración de los diferentes laboratorios.

Gracias al desarrollo de este proyecto, se pretende facilitar el trabajo a los profesores y técnicos de la Facultad de Química a la hora de trabajar, ya que, a partir de su uso ya no será necesario consultar físicamente los almacenes de productos. Mediante la aplicación web o su teléfono móvil se podrá consultar y reconocer los productos que se necesitan para dar una clase, así como también permitirá a los técnicos de laboratorio saber de una manera más rápida y clara cuando es necesario pedir más suministros a un fabricante porque se está llegando al límite de existencias en un laboratorio.

El proyecto se ha desarrollado teniendo en cuenta posibles ampliaciones, ya que se prevé su aplicación a un terreno más amplio a medida que se vaya escalando su magnitud.

En el presente documento se detalla el proceso de análisis, diseño, planificación e implementación de la plataforma web comentada.

Abstract (Català)

El projecte documentat que es desenvoluparà en aquest Treball Fi de Grau consisteix en un sistema de gestió mitjançant una aplicació web de l'inventari de productes químics de la Facultat de Química. El projecte consisteix en el disseny e implementació d'una aplicació web que s'adapti a les necessitats de la Facultat de Química i permeti gestionar diferents inventaris de productes amb diversos nivells de jerarquia de producte, diferents nivells d'accés, modificació i consulta mitjançant la plataforma web.

Mitjançant aquesta eina, es mantindrà un control i és gestionaran els productes que hi ha en diferents magatzems, juntament amb les seves característiques i relacions, així com l'administració dels diferents laboratoris.

Gràcies al desenvolupament d'aquest projecte, és pretén facilitar el treball als professors i tècnics de la Facultat de Química a l'hora de treballar, ja que, a partir de la seva utilització ja no serà necessari consultar físicament els magatzems de productes. Mitjançant l'aplicació web o el seu telèfon mòbil es podran consultar i reconèixer els productes necessaris per donar una classe, així com permetrà als tècnics de laboratori saber d'una manera molt més ràpida i clara quan és necessari demanar més productes a un fabricant perquè s'està arribant al límit d'existències en un laboratori.

El projecte s'ha desenvolupat tenint en compte les possibles ampliacions, ja que, es preveu la seva aplicació en un terreny més ample a mesura que es vagi escalant la seva magnitud.

En el present document es detalla el procés d'anàlisi, disseny, planificació i implementació de la plataforma web comentada.

Abstract (English)

The following project is a management system using a web application for the inventory of chemistry of the Faculty of Chemistry. The project is the design and development of a web application that fits the necessities of the Faculty of Chemistry and allow them manage their inventories with diferents levels of product hierarchy, levels of acces, modification and search.

Using this tool, the Faculty of Chemistry can check all the products, with all their characteristics and relationships, as well as the administration of the diferents laboratories.

The development of this project will make the work more easy for the teachers and the technicians of the Faculty. With this web application will no longer be necessary physically check the stores, you can do it with this web application. You will be able to recognize the products that you need to teach a class, as well as allow lab technicians know more quickly when to seek for more products when the laboratory is reaching the stock limit.

The development of this project has been developed taking into account possible extensios as they are intended to be applied as the project intensify his magnitud.

In this document we will se the process of analisis, design, planification and development of the project.

1. Introducción

Toda empresa u organización que se dedique de alguna manera u otra a la producción y/o manipulación de productos debe, de alguna manera, contabilizar e identificar las diferentes existencias que tienen en sus almacenes. Para poder realizar estas tareas, es necesario contar con un inventario de productos.

Que es un inventario de productos? Es el conjunto de productos almacenados en espera de su ulterior empleo, para satisfacer una demanda futura. En ellos deben aparecer las relaciones detalladas, ordenadas y valoradas de los elementos.

En nuestro caso no sólo nos encontramos con un inventario de productos, sino que tenemos el caso particular de un inventario de productos químicos. Esta peculiaridad añade diferentes características de seguridad necesarias en un inventario de este tipo, como pueden ser la cantidad y la naturaleza de cada uno de los productos y residuos, clasificación según su naturaleza (inflamables, corrosivos, tóxicos...), evaluaciones de riesgos, realización de tareas preventivas del laboratorio etc.

1.1. Inventario de productos químicos

Los productos químicos que encontramos en estos inventarios no son simples productos como entidad, ya que un producto puede ser una mezcla de una o varias sustancias químicas.

Durante el diseño y posterior desarrollo de este proyecto se han tenido en cuenta las siguientes características y propiedades químicas:

1.1.1. Sustancias Químicas

Por sustancia entendemos una especie de materia homogénea de composición química definida, sin importar su procedencia. Es una forma de materia que tiene una composición química constante y propiedades características. No se puede separar en componentes por métodos de separación física, es decir, sin romper los enlaces químicos. Pueden encontrarse en formas sólidas, líquidas, gaseosas y plasmáticas. Estas pueden cambiar entre las fases de la materia mediante cambios de temperatura o presión. Debido a esto las reacciones químicas convierten una sustancia química en otra.

Hay que diferenciar las sustancias químicas "puras", que es como normalmente se suele llamar a las sustancias, de las mezclas. Las mezclas contienen más de una sustancia con

diferentes niveles de concentración y no tienen una composición fija. Es un sistema formado por dos o más componentes, pero no combinados químicamente. En una mezcla no ocurre una reacción química y cada uno de sus componentes mantiene su identidad y propiedades químicas. La combinación física que retiene sus identidades y que se mezclan entre ellas pueden formar, según sea el caso aleaciones, soluciones, suspensiones y coloides.

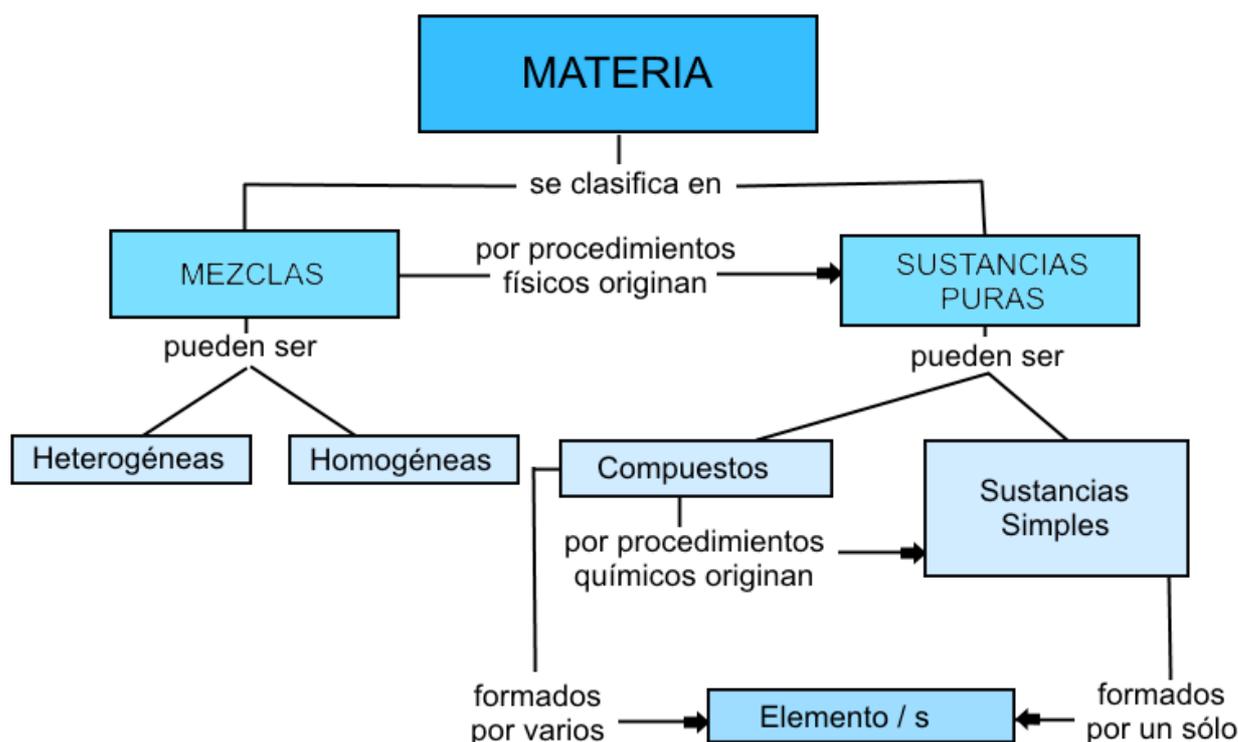


Figura 1: Sustancias Puras y Mezclas [Fuente 5]

1.1.2. Número de registro CAS

Debido a la complejidad que presentan las sustancias químicas y el gran número de combinaciones que pueden formar unas con otras, en química se utiliza el número de registro CAS para identificarlas.

El número de registro CAS, es un identificador numérico único (como un ID) designado a una única sustancia química. Este código no tiene ningún significado químico per se, sino que enlaza con una gran cantidad de información acerca de la sustancia química específica. Puede contener hasta 10 dígitos, divididos por guiones en tres partes. El dígito

del extremo es un dígito de control para verificar la validez y originalidad de todo el número. Este número CAS se convirtió en un estándar a nivel mundial, ya que, no depende de ningún sistema de nomenclatura química y proporciona un vínculo fiable entre los diferentes términos que se usan para describir las sustancias y compuestos químicos. Este hace posible que aunque una misma sustancia se pueda escribir de muchas maneras diferentes: nombres genéricos, nombres comerciales, nombres sistemáticos, fórmula molecular, por su estructura, siempre podremos identificar una única y específica sustancia independientemente del número de maneras posibles que puede ser descrita.

El CAS (*Chemical Abstracts Service*) es una división de la *American Chemical Society* y es la autoridad mundial respecto a la información química. Esta es la única organización en el mundo cuyos objetivos son hallar, recopilar y organizar toda la información sobre sustancias químicas de forma pública. Formado por un equipo de científicos que comisionan y controlan la calidad de estas bases de datos, las cuales son reconocidas por las compañías farmacéuticas, químicas, universidades, organizaciones gubernamentales y oficinas de patentes de todo el mundo.

CAS Registry es el registro que tiene el CAS (*Chemical Abstracts Service*) y contiene una amplia variedad de sustancias, incluyendo la mayor colección mundial de compuestos orgánicos, inorgánicos, organometálicos, elementos, isotopos, partículas nucleares, proteínas y ácidos nucleicos, polímeros. Este registro va aumentando considerablemente de número cada día (alrededor de 15.000 sustancias nuevas diarias).

Chemical Structure

Registry Number: 58-08-2

Formula: C₈H₁₀N₄O₂

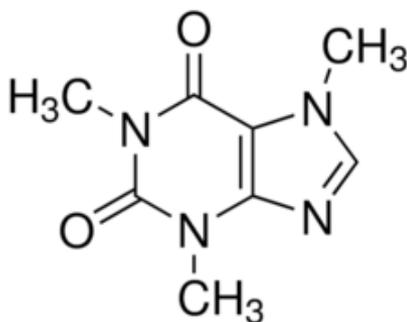


Figura 2. Ejemplo CAS y Formula (cafeína) [Fuente 7]

1.1.3. Seguridad

Como nos encontramos trabajando con sustancias químicas es necesario hablar de la seguridad vinculada a cada una de estas. Todo líquido, gas o sólido que supone un riesgo para la salud o seguridad de las personas debe de estar debidamente comunicado. Debido a que no todas las sustancias tienen el mayor grado de peligrosidad, cada producto tiene asociadas frases de seguridad, pictogramas y una ficha de seguridad.

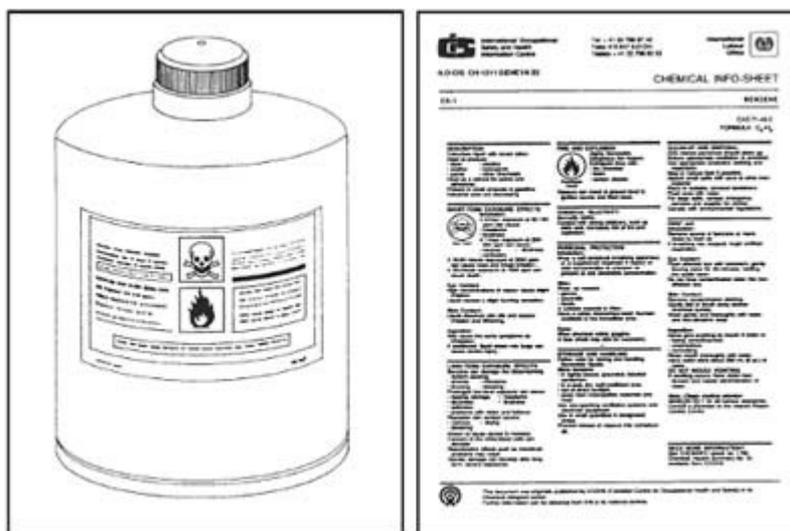


Figura 3. Seguridad en un Producto Químico [Fuente 8]

Encontramos diferentes tipos de frases de seguridad (frases S, R, P, H y EUH). Cada uno de estos conjuntos numerados de frases son usados para indicar consejos de utilización y prudencia, así como para evitar los efectos adversos causados por la exposición a una sustancia o mezcla peligrosa durante su uso o eliminación.

Por otra parte nos encontramos con los pictogramas de seguridad, estos son símbolos que muestran gráficamente el nivel de peligrosidad de la sustancia etiquetada.

Finalmente nos encontramos con las Fichas de Seguridad, también conocidas como FDS, fichas de datos de seguridad o MSDS. Son unos documentos que deben proporcionar gratuitamente las empresas que se dedican a la comercialización de sustancias químicas. Estos documentos consisten en una ficha en la cual se especifican 16 apartados relacionados con las recomendaciones para un uso seguro de las sustancias químicas:

- 1.- Identificación de la sustancia o preparado y de la sociedad o empresa responsable de su comercialización.
- 2.- Composición/Información sobre los componentes.
- 3.- Identificación de los peligros.
- 4.- Primeros auxilios.
- 5.- Medidas de lucha contra incendios.
- 6.- Medidas que deben tomarse en caso de vertido accidental.
- 7.- Manipulación y almacenamiento.
- 8.- Controles de exposición / protección individual.
- 9.- Propiedades físicas y químicas.
- 10.- Estabilidad y reactividad.
- 11.- Información toxicológica.
- 12.- Informaciones ecológicas.
- 13.- Consideraciones relativas a la eliminación.
- 14.- Información relativa al transporte.
- 15.- Información reglamentaria.
- 16.- Otras informaciones.

Mediante estos puntos se pretende informar sobre los riesgos y peligros del producto además de formar al usuario del producto en cuanto al comportamiento y las características del producto, su correcta utilización, los controles de exposición, los sistemas de protección que deben utilizarse en caso de que el control no fuera del todo eficaz o en caso de emergencia, las acciones que deben ponerse en marcha en caso de accidente.

El conjunto de estas características hace posible una gran prevención y seguridad necesaria para la manipulación y el uso de productos químicos.

1.2. Motivación

La motivación de la que nace este trabajo fin de grado se podría dividir en diversos puntos.

El primero de todos es la idea del proyecto en si, que surge de la necesidad de la Facultad de Química de implantar un sistema de gestión moderno que permita gestionar, consultar y modificar el inventario de productos químicos de la facultad mediante una aplicación web.

Esta necesidad me ofrece la oportunidad de aplicar todos los conocimientos adquiridos durante el Grado de Ingeniería Informática en un proyecto de mayor tamaño desarrollado desde cero. Diseñando y solucionando un problema de un cliente mediante una aplicación web creada a medida y aplicada en un entorno de producción real.

Y finalmente, gracias a la libertad que se ofrece, se me brinda la oportunidad de poder investigar mas el mundo del desarrollo web y ampliar mis conocimientos utilizando un framework PHP relativamente nuevo y con mucho potencial como es Laravel y aplicarlo a un proyecto para un cliente real.

1.3. Objetivos

El objetivo de esta aplicación es la implementación de un sistema de gestión mediante una aplicación web que permita a la Facultad de Química de la Universidad de Barcelona gestionar el inventario de productos químicos de esta.

Se pretende ayudar y facilitar a los usuarios de la plataforma la gestión y consulta del material de laboratorio de una manera mucho mas rápida respecto a los medios tradicionales que cuenta actualmente la Facultad y además tiene que ser accesible desde cualquier dispositivo móvil u ordenador.

Se contará con un generador interno de códigos QR que permitirá el etiquetado de los productos del inventario. Además a cada producto se le asignará un identificador único mediante un código HASH, que asegurará poder diferenciarlo de los productos parecidos.

La aplicación contará con diferentes niveles de administración. Una sección para profesores, otra sección para técnicos de laboratorio y finalmente una sección para administradores.

Los profesores podrán acceder a la aplicación mediante su usuario y contraseña de la UB. Una vez dentro de la aplicación podrán utilizar el buscador de productos por diferentes

criterios de búsqueda (nombre o fórmula química de una sustancia, CAS de la sustancia o código HASH del producto) y en los diferentes laboratorios de la facultad. Una vez encontrado o encontrados los productos que se deseaban al realizar la búsqueda, podrán consultar todos sus detalles, así como la localización, disponibilidad, estado etc. del producto o incluso imprimir los resultados de la búsqueda realizada. También se contará con la posibilidad de reconocer un producto etiquetado mediante un código QR (generado anteriormente por la aplicación) y visualizarlo en su dispositivo móvil (mediante una aplicación móvil) o en su ordenador (mediante un lector de QR conectado por bluetooth) los detalles del producto.

Respecto a la sección del técnico de laboratorio, este accederá a la aplicación mediante un usuario y contraseña propio del sistema. Tendrá la capacidad de crear y editar sustancias, podrá añadir una imagen y un fichero CDX que corresponda a la sustancia y asignarle diferentes características. También podrá crear y editar contenidos, que es donde se crearan las mezclas de sustancias con su diferente concentración y estados físicos de la materia. Deberá poder crear y editar nuevos contenedores (recipientes) y nuevas localizaciones en caso de que sea necesario. Por último, el técnico de laboratorio, será el principal usuario que añadirá y gestionará los productos del inventario, por lo que podrá crearlos y editarlos. Respecto a los productos podrá cambiar su estado, retirarlos, el uso en el que se encuentra, el fabricante, la fecha de caducidad, la fecha en que se ha retirado si es que se ha retirado, podrá reasignar diferentes sustancias a ese producto y tendrá en cuenta la seguridad que envuelve al producto que trate (frases de seguridad, FDS, pictogramas y palabras de advertencia), así como el laboratorio donde se encuentra.

Finalmente tenemos al usuario Administrador que posee todas las capacidades del técnico de laboratorio, además de la capacidad de eliminar cualquiera de los elementos mencionados con anterioridad y la gestión de las diferentes características disponibles para las sustancias y los diferentes pictogramas y frases de seguridad que pueden asociarse a un producto concreto. Al igual que el técnico de laboratorio, accederá a la aplicación mediante un usuario y contraseña propios del sistema.

Por último, destacar el objetivo de crear una plataforma modular y escalable a distintos laboratorios y facultades, ya que este proyecto se aplica a una sola facultad con diversos

laboratorios pero se ha diseñado para poder ampliarse y usarse en diferentes facultades y de esta manera poder gestionar diversos inventarios de productos.

1.4. Distribución de la Memoria

En esta memoria se incluirán los siguientes puntos respecto al proyecto tratado:

- Capítulo 1: Capítulo de introducción donde se detallan nociones básicas, la motivación del proyecto y sus objetivos y como esta organizada la memoria del proyecto.
- Capítulo 2: Se detallan los requisitos del sistema.
- Capítulo 3: Diseño de la base de datos y de la jerarquía de los productos.
- Capítulo 4: Tecnologías utilizadas en el proyecto.
- Capítulo 5: Laravel y sus características mas importantes.
- Capítulo 6: Como esta construido el proyecto.
- Capítulo 7: Planificación que se ha realizado.
- Capítulo 8: Los diferentes tests con los que se ha probado la aplicación.
- Capítulo 9: Conclusiones y trabajo futuro.
- Bibliografía: Citaciones de la documentación que se ha consultado.
- Anexo: Contenido adicional y manuales de uso.

2. Análisis de requerimientos

2.1. Análisis de requerimientos y funcionalidades

Para el desarrollo de este proyecto se analizaron previamente una serie de requisitos que tenían que cumplirse y una serie de funcionalidades que se tenían que aportar. En este apartado los comentaremos.

Estos requisitos y funcionalidades se han ido cumpliendo conforme el proyecto iba avanzando e iba tomando forma.

El requisito principal de la aplicación era poder buscar entre los diferentes productos almacenados en los inventarios de la Facultad de Química y permitir a los técnicos de laboratorio gestionarlos de una manera fluida. Además se querían tener todos los datos de los productos en un mismo lugar facilitando así la búsqueda de información cuando fuera necesaria.

2.2. Requisitos funcionales

Para los requisitos funcionales se han separado en 4 grandes grupos en función de los diferentes tipos de usuarios a los que va enfocada la aplicación web.

- Anónimo: Usuario que accede al login de la UB.
- Profesor: Usuario identificado como profesor de la Facultad de Química.
- Técnico de laboratorio: Usuario identificado como técnico de laboratorio en la aplicación web.
- Administrador: Usuario identificado como administrador en la aplicación web.

A continuación se detallaran los diferentes casos de uso para cada tipo de usuario comentado en el punto anterior. En el anexo del presente documento se pueden consultar con mas detalle los diferentes casos de uso textuales que podemos encontrar en la aplicación.

2.2.1. Casos de uso para el usuario Anónimo

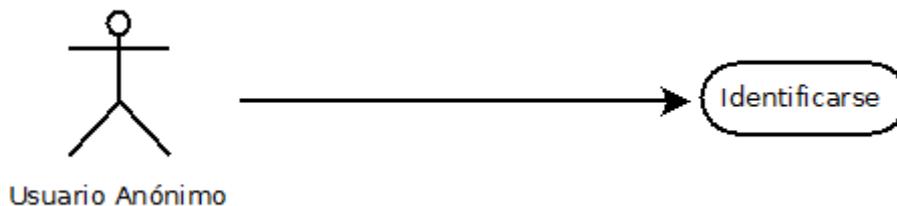


Figura 4. Diagrama de casos de uso del usuario anónimo

Consideraríamos a este tipo de usuario anónimo como cualquier usuario que a priori deseara entrar en la aplicación web, ya que, sólo podrá identificarse mediante el login de la UB y no podrá realizar ninguna otra acción.

2.2.2. Casos de uso para el usuario Profesor

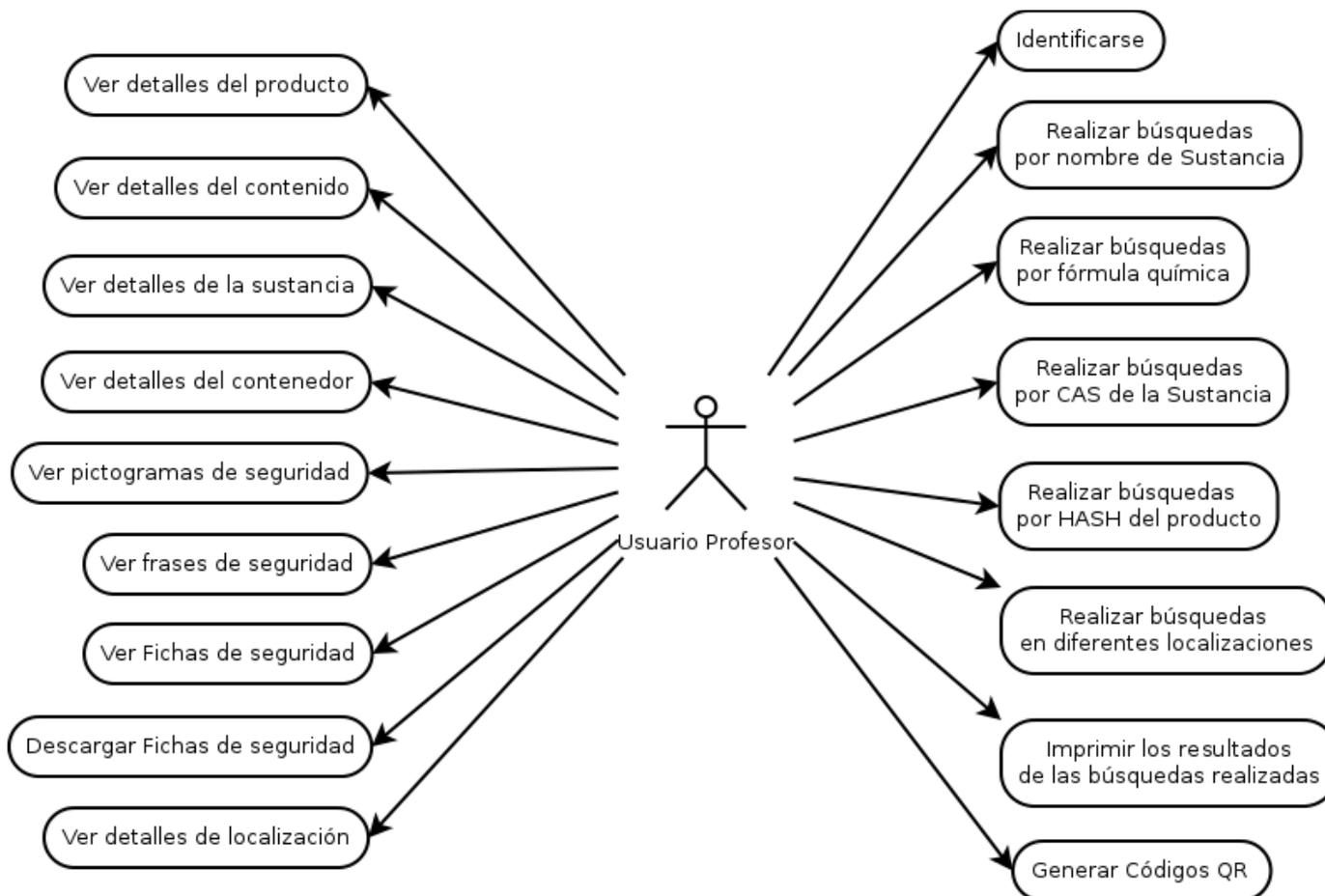


Figura 5. Diagrama de casos de uso del usuario profesor

Un usuario profesor es aquel que se identifica mediante su usuario y contraseña de la UB en la aplicación. Una vez en el sistema puede realizar todas las acciones de consulta para buscar cualquier producto introducido en el sistema. Una vez visualizados los resultados de la búsqueda realizada, el usuario profesor puede imprimir estos resultados como un archivo PDF o puede consultar mas a fondo los detalles de un producto concreto.

Dentro de los detalles de producto, podrá visualizar todos sus datos relacionados con el producto, es decir, las sustancias que lo componen, el contenido donde esta almacenado, la seguridad asociada al producto, en que almacén se encuentra etc.

También aparece una opción de un login adicional, que permite al usuario profesor identificarse como un técnico de laboratorio o administrador del sistema y acceder a las funcionalidades exclusivas de que disponen, en caso de que tenga acceso.

2.2.3. Casos de uso para el usuario Técnico de Laboratorio

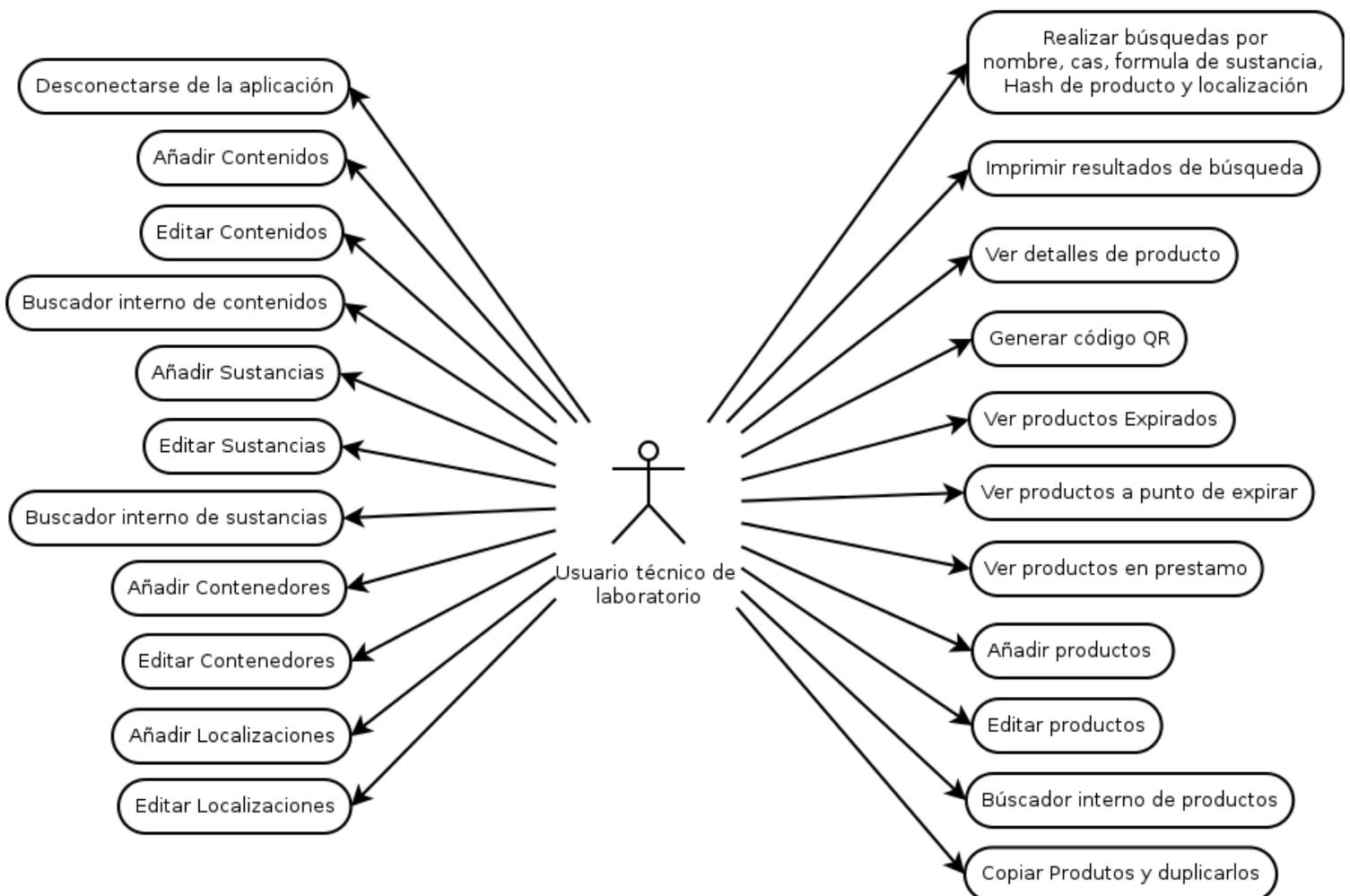


Figura 6. Diagrama de casos de uso del técnico de laboratorio

El usuario técnico de laboratorio tiene acceso a realizar las mismas acciones básicas que el usuario profesor, adicionalmente cuenta la posibilidad consultar las listas de productos expirados, los que están a punto de expirar en los próximos 10 días y los productos que se encuentran en préstamo. De esta manera puede llevar un control de los productos del inventario y así realizar las tareas que se consideren necesarias a partir de los datos obtenidos.

Como nos encontramos ante un usuario de gestión, el técnico de laboratorio tiene la capacidad de editar productos, sustancias, contenidos, contenedores y localizaciones, pudiendo así modificar algunos de sus atributos (no todos, ya que, aquellos sensibles sólo pueden ser accesibles desde un perfil de administración) y características. También cuenta con la funcionalidad de poder copiar productos existentes, facilitando de esta manera la introducción de productos similares, reaprovechando las características y relaciones de los productos introducidos con anterioridad y aumentando la velocidad de gestión del inventario de productos. Para facilitar las tareas de edición de los elementos del inventario, cuentan con unos buscadores internos que permiten reducir el cerco de búsqueda de la aplicación hacia las partes específicas, centrándose así en las sustancias, contenidos y productos.

Finalmente cuenta con una opción para desconectarse de su usuario técnico de laboratorio de la aplicación y así poder cerrar la sesión en el sistema.

2.2.4. Casos de uso para el usuario Administrador



Figura 7. Diagrama de casos de uso del administrador

El usuario administrador tiene acceso a todas las funcionalidades de la aplicación por lo que incluye las funciones del usuario profesor y del usuario técnico de laboratorio, además de las únicas del administrador. Estas son principalmente la gestión del apartado sensible, tales como las frases de seguridad, los pictogramas y las características de las sustancias. Estos apartados pueden ser editados, eliminados y agregados por el usuario o usuarios administradores del sistema.

Debido a que el usuario de administrador es el que controla el sistema del inventario y la eliminación de productos es un asunto sensible, sólo este usuario tiene la capacidad de eliminar de la plataforma los diferentes productos, sustancias, contenidos, contenedores, localizaciones etc. Además de la característica de poder editar una fecha de retirada,

generada automáticamente por la aplicación cuando se retira un producto, en caso de que haya se haya producido algún error o equivocación.

2.3. Requisitos no funcionales

- **Accesible:** La aplicación tiene que ser accesible a través de diversos dispositivos, tanto móviles o tablets como desde ordenadores. Para ello se utilizará “*Responsive web design*” que permita adaptar el diseño de la web a cada uno de estos dispositivos sin tener que realizar una aplicación específica para cada uno de ellos.
- **Ampliable:** La plataforma creada tiene que ser ampliable a mayor escala partiendo como base la posterior ampliación que sufrirá al añadir mas laboratorios y almacenes y un posterior sistema de control de prestamos entre ellos.
- **Fluida:** La aplicación tiene que ser lo más ligera posible en cuanto a la gestión de base de datos de los inventarios, permitiendo importar, exportar, retroceder etc. de una manera rápida y sencilla.

3. Diseño del modelo de datos

3.1. Jerarquía de productos en la aplicación

Este sistema de gestión web cuenta con una jerarquía de productos químicos particular. Esta jerarquía se ha implementado de esta forma concreta debido a las características comentadas anteriormente sobre las sustancias químicas y su capacidad para combinarlas unas con otras creando otras sustancias compuestas o mezclas.

La jerarquía de productos en este proyecto es la siguiente:

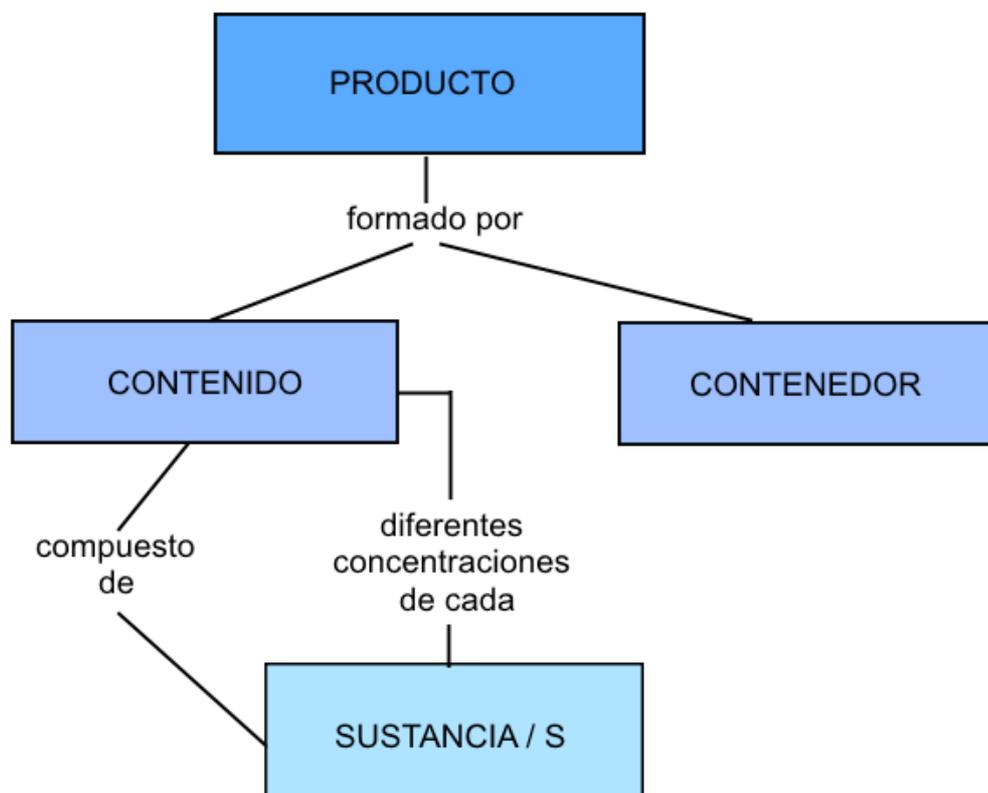


Figura 8. Diagrama de la jerarquía de productos

Como se puede comprobar en la ilustración anterior vemos que un producto del inventario se considera una unidad de almacenamiento con una serie de características asociadas, formado por un contenido y un contenedor. Por contenedor entendemos el recipiente o envase donde se almacena el contenido. Y por contenido entendemos la especie, mezcla de sustancias o sustancia en una concentración determinada.

3.2. Modelo Entidad / Relación (ER)

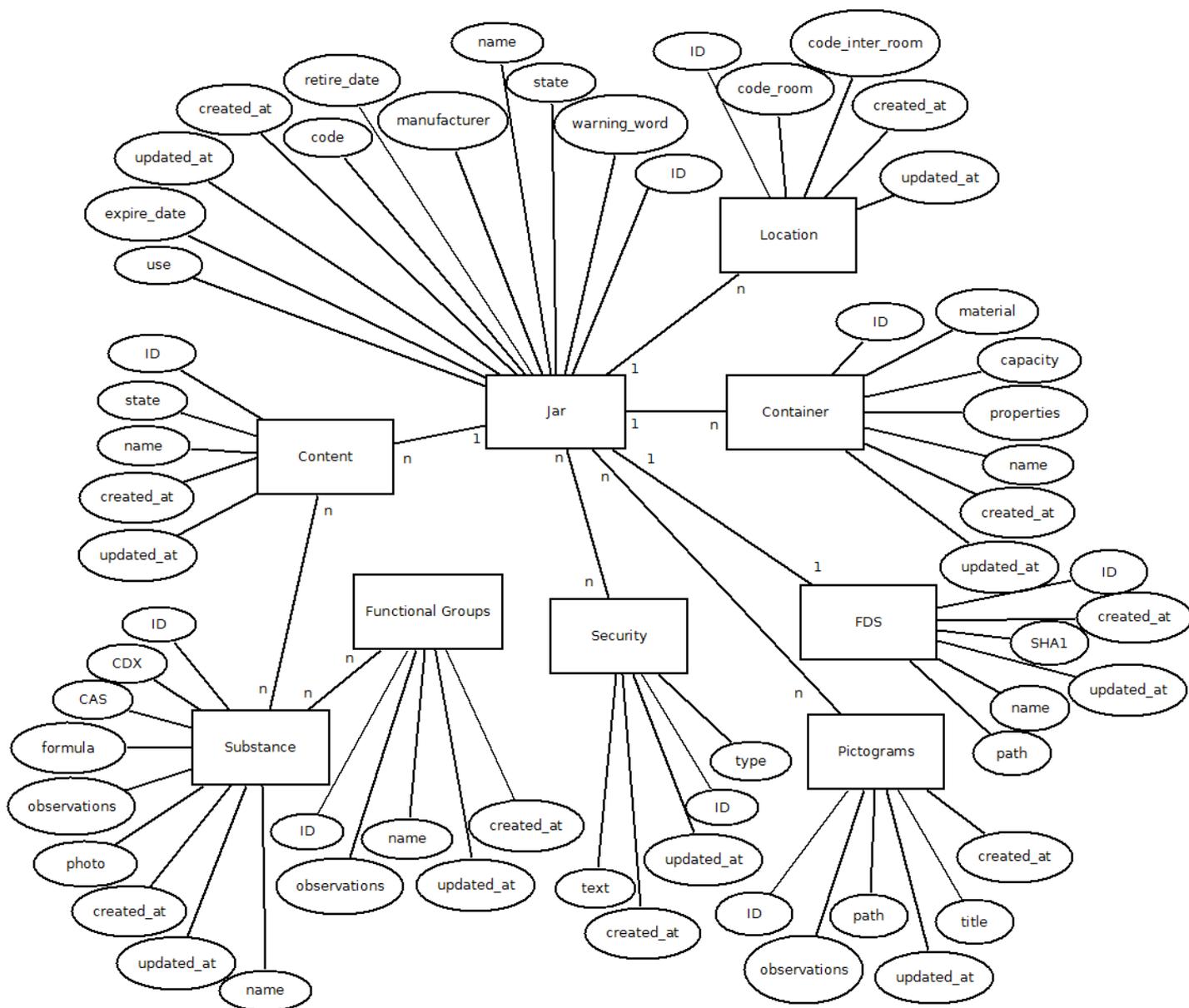


Figura 9. Modelo ER

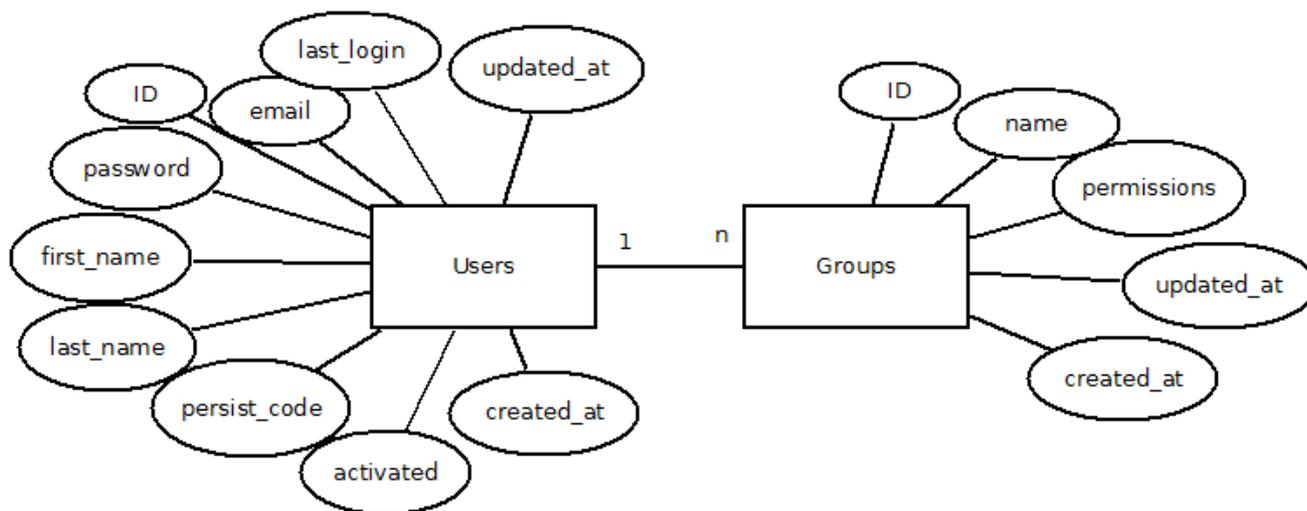


Figura 10. Modelo ER Usuarios

En la ilustración número 9 podemos observar el modelo ER que representa las relaciones del modelo de datos. En la figura número 10 que se puede observar a continuación, se muestra el modelo ER de los usuarios y la relación entre sus datos.

3.3. Modelo de Datos

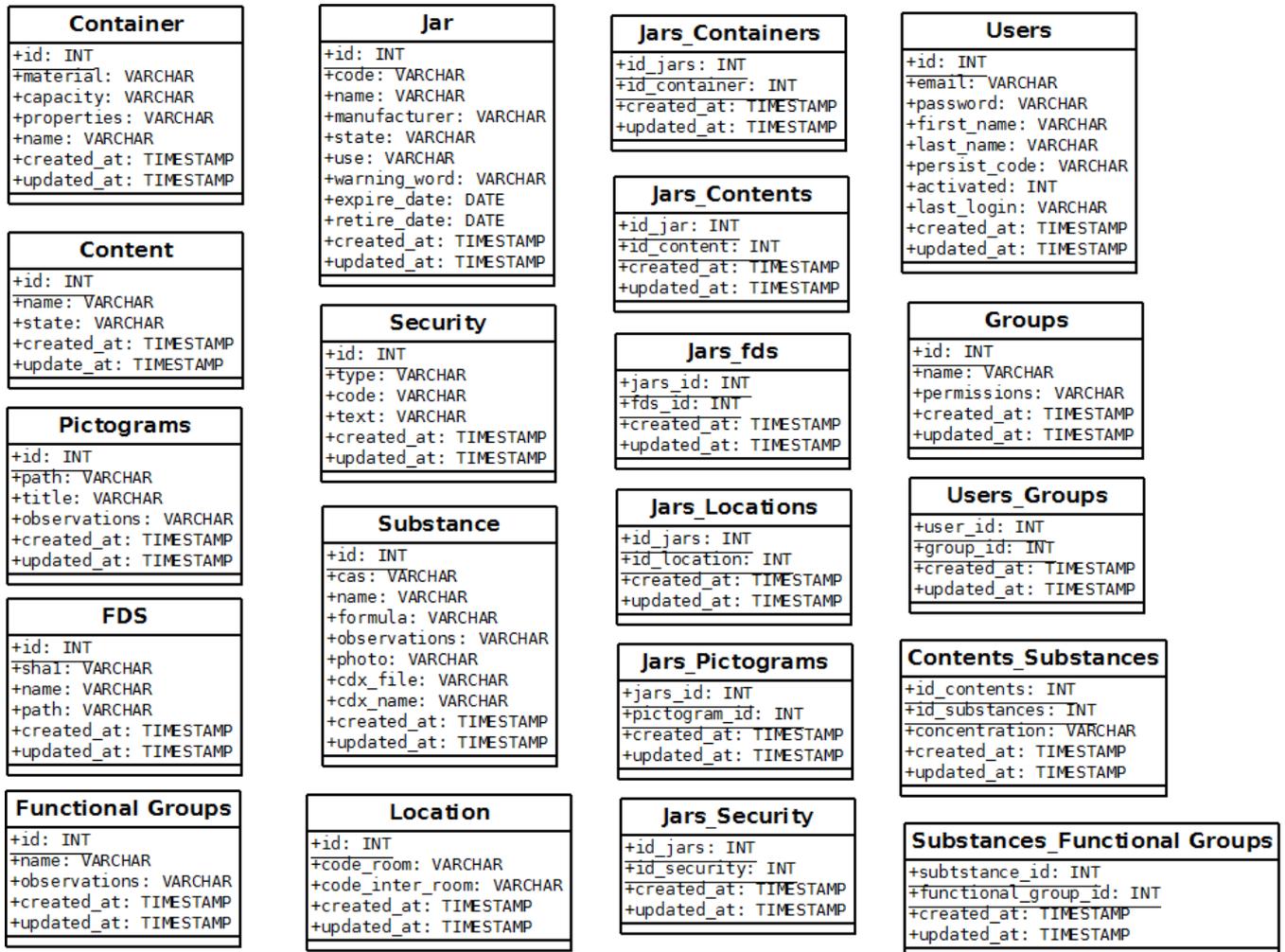


Figura 11. Modelo ER Usuarios

En la figura número 11 podemos visualizar las diferentes tablas que componen la base de datos junto con los atributos de cada una de ellas. Podemos observar 3 tipos de tablas:

- Las tablas de las entidades, que almacenan los diferentes atributos para cada una de ellas.
- Las tablas de relaciones, que almacenan las diferentes relaciones entre las entidades del proyecto.
- Las tablas de usuario, que vienen a ser una combinación de las dos anteriores pero específicas para los usuarios de la aplicación y a los grupos de administración a los que pertenecen.

4. Tecnología utilizada

4.1. Servidor LAMP – Linux / Apache / MySQL / PHP

Para el desarrollo de esta aplicación web se utilizó un servidor bajo una arquitectura LAMP (Linux, Apache, MySQL y PHP) [9].

Este servidor de desarrollo se implementó mediante una máquina virtual de VMWare [10]. En un primer momento se eligió la distribución de Ubuntu 10.04 LTS [11] por su ligereza pero debido a que la aplicación se ha desarrollado para la Universidad de Barcelona y esta tiene un servicio de Hosting Web propio [13] en las fases iniciales del desarrollo se optó por crear un entorno de desarrollo lo más similar posible al entorno de producción donde finalmente se alojaría la plataforma, y por este motivo se migró hacia una máquina virtual VMWare con una distribución de Ubuntu 12.04 LTS [12]. Se ha buscado una distribución que contara con *Long Term Support (LTS)* previniendo el soporte a largo plazo necesario para las posibles ampliaciones de la plataforma y asegurar el correcto mantenimiento de esta. Concretamente Ubuntu 12.04 tiene soporte asegurado hasta abril de 2017.

Requisitos Servidor UB:

- Sistema Operativo Linux 3.0
- Servidor Web Apache 2.2.x
- PHP 5.x

Servicios instalados en el servidor virtual de desarrollo:

- Apache 2.2.22
- PHP 5.3
- MySQL 5.5

Para llevar a cabo el desarrollo junto con el servidor virtual se creó una unidad de red virtual que comunica el directorio del proyecto con el servidor virtual. De esta manera se permite desarrollar fuera de la máquina virtual permitiendo mayor realismo al entorno de desarrollo. Esto se realizó a través del propio VMWare mediante la configuración de sus *Shared Folders*.

4.2. Composer

Composer [14] es una herramienta que permite gestionar dependencias en PHP. Nos permite declarar las dependencias que nuestro proyecto necesita y las instala en el proyecto por nosotros.

No hay que confundir Composer con un gestor de paquetes, gestiona paquetes y librerías sí, pero siempre lo hace a nivel de proyecto (por ejemplo en el directorio vendor de nuestro proyecto). Por defecto nunca instalará nada globalmente. Es por esto que nos referimos hacia el como un gestor de dependencias.

Composer se ejecuta mediante la línea de comandos e instala dependencias (librerías) para la aplicación. Además permite al usuario instalar aplicaciones PHP disponibles en "Packagist" [15] que es el principal repositorio de paquetes.

Los problemas que Composer soluciona son los siguientes:

- Tenemos un proyecto que depende de un número de librerías.
- Algunas de estas librerías dependen de otras librerías.
- En nuestro proyecto, declaramos las cosas de las que dependemos.
- Composer encuentra que versiones de los paquetes necesita instalar y los instala, es decir, los descarga en nuestro proyecto para poderlos utilizar.

4.3. Framework PHP – Laravel

Un framework es una estructura conceptual y tecnológica de soporte definido que puede servir de base para la organización y el desarrollo de software. Básicamente un framework hace mas fácil para el programador el poder desarrollar una aplicación web.

La gran mayoría de aplicaciones tienen en común una serie de funcionalidades como gestionar las sesiones, la validación de datos etc y un framework permite evitar tener que reescribir cada una de estas funcionalidades cada vez que se crea un proyecto nuevo.

La creación de la plataforma web propuesta para este trabajo fin de grado se basa en el framework PHP Laravel. Laravel es un framework PHP de código abierto que permite desarrollar aplicaciones y servicios web mediante PHP 5. Se caracteriza por su filosofía de

código limpio y expresivo, permitiendo así que nos ocupemos de la parte divertida del proyecto que no es otra que la creación y desarrollo de este.

Laravel se profundiza en el siguiente capítulo, **5. Laravel a fondo**.

4.4. Framework HTML & CSS – Bootstrap

Bootstrap es el framework HTML, CSS y JS mas popular para desarrollar proyectos web que cumplan un diseño *responsive* [16] y la tendencia *mobile first* [17]. Fue creado por el equipo de desarrolladores de Twitter cuando buscaban una forma de acelerar sus proyectos y compartir bienes y patrones de diseño dentro de su propia compañía. En Agosto de 2011 se convirtió en un proyecto de código abierto y a mediados de Febrero de 2012 fue el proyecto mas popular de Github.

Este éxito de Bootstrap se debe a que actualmente cualquier web no es solo accesible desde un ordenador, sino que actualmente se puede acceder y navegar por ella con una variedad inmensa de dispositivos móviles, tales como tablets o smartphones. Debido a esto nació en la comunidad de desarrolladores web la necesidad de implementar soluciones para que las webs pudieran interactuar con estos dispositivos. Es en este momento donde nacen las tendencias *Responsive web design* y *Mobile First*.



Figura 12. Responsive Web Design

Bootstrap a partir de diferentes archivos HTML, CSS y Javascript incluidos en su framework adapta estas tendencias y proporciona una serie de ventajas a los desarrolladores:

- *Mobile First*: Todo el framework está desarrollado teniendo en cuenta la filosofía mobile first que es la más recomendable para el desarrollo de sitios web responsive.
- *Mantenimiento*: Cuenta con el mantenimiento y las actualizaciones del equipo de Twitter, por lo que está en constante mejora.
- Utiliza componentes creados por la comunidad web, tales como HTML5, CSS3, Normalize.css, JQuery, Github etc.
- *Grid Responsive*: El grid de bootstrap es 100% responsive y te permite pensar en todas tus vistas para ofrecer la mejor experiencia posible al usuario.
- *Compatibilidad*: Está diseñado para funcionar con todos los navegadores modernos, incluyendo soporte para IE8.
- *LESS*: Bootstrap está optimizado para LESS, un potente preprocesador que funciona con lenguaje de programación y permite el uso de variables, funciones entre otras para acelerar y enriquecer los estilos de los sitios web.
- *Font Icons*: Utilizan el font icons vectoriales para mejorar el rendimiento.
- *Default Theme*: El tema que ofrece por defecto está bastante optimizado y permite modificarlo o crear tu propio tema a partir de este.

La aplicación web creada se basó en el framework de Bootstrap para adaptar su diseño a todos los dispositivos móviles, cosa que facilitó enormemente la creación de las vistas de la aplicación ya que automáticamente las adapta al tamaño de pantalla en el que se encuentran, permitiendo el desarrollo con una sola hoja de estilos en vez de tener que crear una para cada tipo de dispositivo.

4.5. Repositorio de código – Github

En el corazón de Github encontramos Git, un proyecto open source empezado por el mismo creador de Linux. Git es como otros sistemas de control de versiones, gestiona y almacena diferentes revisiones de proyectos. A diferencia de otros predecesores, que aplicaban los cambios que se hacían en un proyecto directamente en el repositorio central, Git copia el proyecto entero a tu sistema. Esto permite que después de hacer los cambios necesarios, sea posible hacer un “check in” con el servidor central. De esta manera no es necesario tener que conectar con el servidor cada vez que se realiza un cambio.

Github no es mas que un servicio de hosting de Git. Este ofrece funcionalidades adicionales como una interfaz web y otras funcionalidades extra como el uso de wikis y herramientas básicas para la gestión de cada proyecto.

Para este proyecto se ha utilizado un repositorio Github para tener control de versiones durante el desarrollo y, gracias a la posibilidad de crear diferentes ramas de desarrollo, se ha creado una rama para el desarrollo local y otra para el código en el servidor de producción. De esta manera se ha podido separar el entorno de desarrollo del de producción de una manera fácil pero a la vez accesible.

4.6. Metodologías Ágiles – Scrum y Kanban

Para la planificación y desarrollo del proyecto Inventario de Química se han utilizado metodologías ágiles. Estas metodologías permiten el desarrollo iterativo e incremental donde los requisitos y soluciones evolucionan mediante la colaboración de un grupo auto organizado y multidisciplinario. Se ha decidido utilizar este tipo de técnicas de gestión de proyectos debido a que los intereses durante el desarrollo para la Facultad de Química se basaban sobretodo en la interacción con el cliente para hacer un software especializado y a medida además de poder ir realizando respuestas a los cambios que fueran necesarios según las circunstancias del proyecto. Como se puede notar, estas características concuerdan en su gran parte con el manifiesto ágil [24].

En el caso de este proyecto se optaron por las siguientes metodologías ágiles, adaptándolas a las necesidades de este:

- **Scrum:** Es un marco de trabajo que nos proporciona una serie de herramientas y roles para, de una forma iterativa, poder ver el progreso y los resultados de un proyecto.

Para tener un seguimiento iterativo de la plataforma se realizaban reuniones semanales donde se repasaban el estado del sprint anterior (sprint review), los requisitos de la aplicación y se formaba un nuevo sprint backlog (lista de tareas para ese sprint).

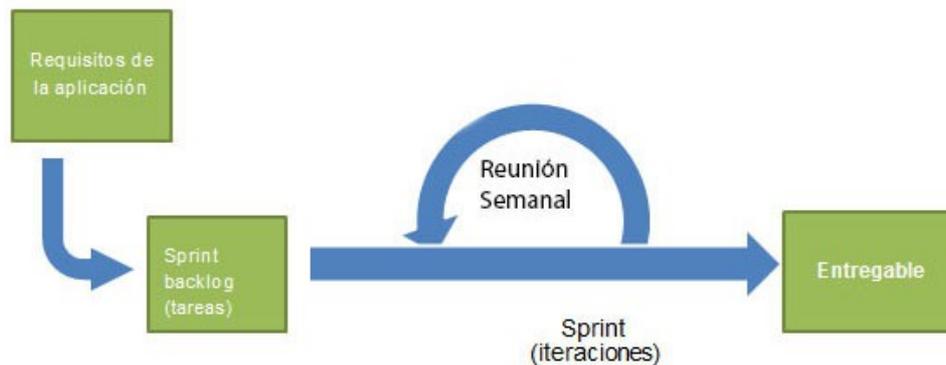


Figura 13. Adaptación del esquema del Scrum aplicado al proyecto

- **Kanban:** Se basa en la idea de dividir el trabajo en secciones y que el trabajo en curso debería limitarse a un número reducido de tareas. Haciendo que solo podamos empezar algo nuevo cuando un bloque de trabajo anterior haya sido finalizado.

En los sprint backlog comentados anteriormente se asignaban las tareas para el siguiente sprint y estas se asignaban a la columna correspondiente del diagrama de kanban que representaba ese sprint. De esta manera se tenía un control de las tareas que se habían planificado y el estado de cada una de ellas.

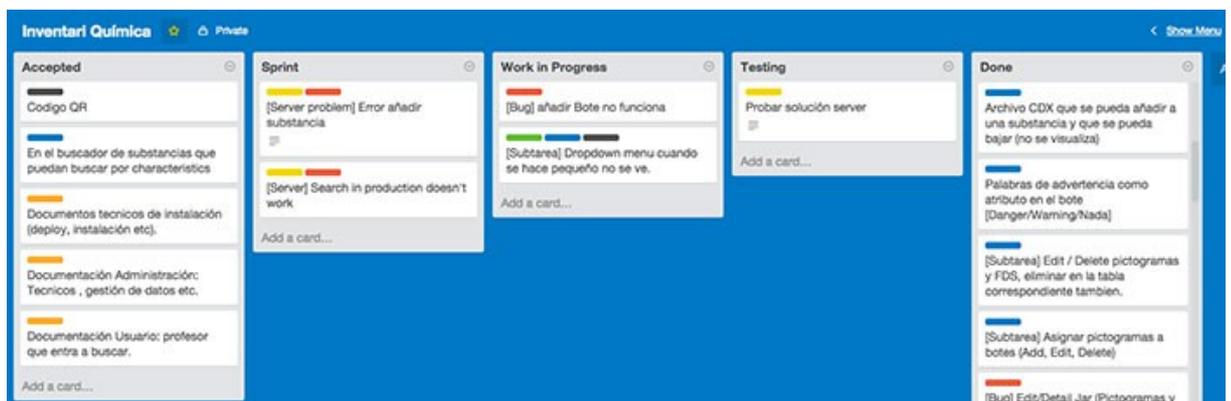


Figura 14. Kanban con diferentes tareas asignadas

5. Laravel a fondo

5.1. Laravel - *The framework for web artisans*

¿Que es Laravel? Laravel es un framework PHP de última generación que se basa en PHP 5.3. Fue creado en 2011 por Taylor Otwell y es famoso por su filosofía de la convención ante la configuración, permitiendo al desarrollador escribir código rápido, limpio y elegante, quitando de en medio lo tedioso del desarrollo de aplicaciones.

Se considera a Laravel un framework para aplicaciones web con una sintaxis elegante y expresiva. Laravel intenta hacer del desarrollo web una experiencia agradable intentando que el desarrollador no se preocupe por las tareas comunes de la mayoría de proyectos webs, tales como la autenticación, el routing, las sesiones y el cacheado, pudiendo centrarse en la creación de la aplicación web en si. Pretende convertir el proceso de desarrollo en algo agradable para el programador sin sacrificar la funcionalidad de la aplicación. Tal y como dicen la página web de Laravel [26] "*Happy developers make the best code*", de esta manera Laravel deja claro la línea que sigue su filosofía.

Nos encontramos ante un framework accesible y aun así poderoso que provee poderosas herramientas para la creación de aplicaciones robustas y de gran tamaño.

Laravel combina lo mejor de lo que se ha visto en otros frameworks, incluyendo frameworks implementados en otros lenguajes de programación. Ha sido influenciado por tales como Ruby on Rails, ASP .NET MVC y Sinatra. Laravel se basa en componentes de Symfony2[30], otro framework PHP mucho mas complejo, permitiendo utilizar los componentes necesarios para aprovechar toda su rentabilidad.

5.2. Frameworks PHP

A la hora de crear la aplicación web, nos encontramos con la siguiente duda: ¿Sobre que framework basar nuestro desarrollo?

Teniendo en cuenta los requisitos mencionados en el apartado anterior se ha realizado un análisis de los frameworks de desarrollo web mas famosos buscando los puntos fuertes que nos pueden interesar y las debilidades que pueden suponer un problema para nuestro interés.



| Ventajas | Inconvenientes |
|-----------------------------|--|
| Sencillo | Recursos no necesarios |
| Buena base de usuarios | Necesarios muchos plugins para tapar carencias |
| Buena documentación | Demasiado básico |
| Curva de aprendizaje ligera | Soporte a PHP 4 |
| Muy rápido | No cuenta con separación modular |

CodeIgniter es un framework PHP muy básico que ofrece una base simple para el desarrollo de aplicaciones. El problema es que es demasiado básico para nuestro caso, ya que una vez nos enfrentamos a nuestro proyecto vemos que hay carencias en cuanto a sus modelos de datos y vemos necesario utilizar mucha parte de nuestro código en el controlador de la aplicación debido a que no cuenta con una separación de módulos, haciéndola una súper clase a cargo de casi todo. También es necesario basarnos en demasiados pluggins externos para suplir algunas carencias importantes. Además por otra parte sigue dando soporte a PHP 4 y esto implica no poder aprovechar al máximo la potencia de un framework basado plenamente en PHP 5.

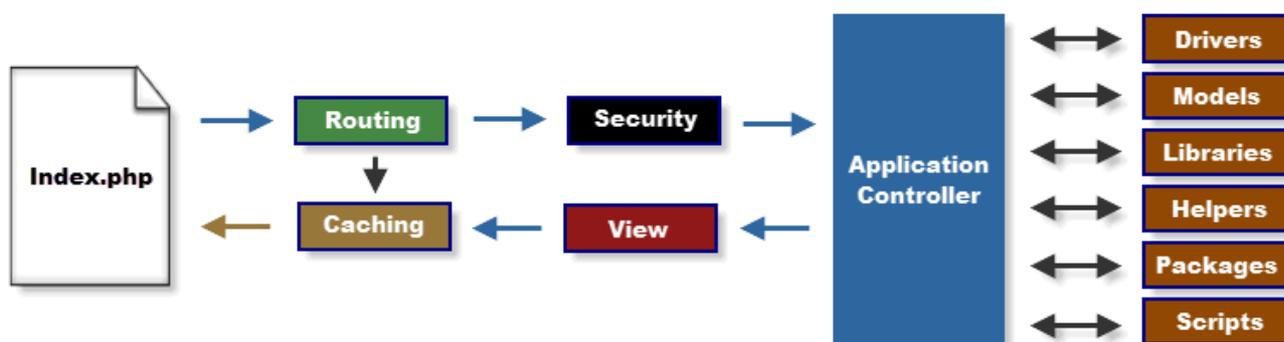


Figura 15. Diagrama del controlador en CodeIgniter



| Ventajas | Inconvenientes |
|--------------------------------|--|
| Estándar de aplicaciones web | Demasiadas opciones y recursos |
| Muy completo | Complejo |
| Robusto | Mas tiempo para desarrollar una aplicación |
| Gran base de usuarios | Demasiado grande para un proyecto no corporativo |
| Excelente documentación | |
| Contiene todos los componentes | |

Cuando hablamos de symfony2, estamos hablando del framework mas robusto que hay actualmente. Esto puede ser una gran opción para una corporación grande que tenga que realizar un proyecto de una envergadura considerable, pero no es nuestro caso. Para el proyecto en el que nos encontramos, si utilizáramos symfony2 estaríamos desperdiciando una gran cantidad de recursos y sería como lo comúnmente llamado “matar una mosca a cañonazos”. Debido a la gran potencia de Symfony2 se necesita un servidor web potente para tener un rendimiento adecuado a las características que incorpora. El servidor que nos ofrece la Universidad de Barcelona es un servidor estándar y por lo tanto no vemos buena opción el uso de Symfony2 en él. Por otra parte la curva de aprendizaje de Symfony2 es la mas elevada que nos encontramos y debido a que no aprovecharíamos todo su potencial se opta por descartar esta opción.



| Ventajas | Inconvenientes |
|--------------------------------|--------------------------------|
| Sencillo | Muy poca información en inglés |
| Buenos hábitos de programación | Fuerte en otros países |

| | |
|------------------------|--|
| Bastante documentación | Poca maduración fuera de república checa |
| Buena base de usuarios | |

Nette tiene un planteamiento bastante parecido a Laravel en cuanto a características pero tiene un gran inconveniente. Debido a su origen tiene muy poca presencia fuera de la república checa y esto hace muy difícil el acceso a más información, creando una barrera de lenguaje bastante importante.



| Ventajas | Inconvenientes |
|----------------------------------|-----------------------------|
| Curva de aprendizaje sencilla | POO no moderna |
| Basado en diferentes componentes | No composer |
| Bastante documentación | Sobreuso de arrays de datos |
| Gran comunidad | No Moderno |

Cake PHP fue uno de los primeros frameworks que salieron a la luz, por este motivo no cuenta con una programación orientada a objetos moderna, ni cuenta con composer. Debido a esto el mantenimiento del código puede ser un gran problema ya que estaríamos estandarizando algo que se intenta reemplazar y mejorar por las corrientes más modernas. Estas faltas hacen que no este a la altura de los frameworks modernos y por lo tanto fue descartado de la lista de frameworks.



| Ventajas | Inconvenientes |
|--------------------|-----------------------------|
| Framework sencillo | Documentación desorganizada |
| Composer | |
| POO moderna | |

Yii es un framework moderno que cuenta con características bastante parecidas a Laravel, el problema que nos encontramos con yii es su documentación. Esta muy desorganizada y es caótica y esto hace que se descarte inmediatamente, ya que al empezar a trabajar con un nuevo framework es muy necesario tener el respaldo de una buena documentación donde poder consultar si hace falta.



| Ventajas | Inconvenientes |
|------------------|--------------------------------------|
| Muy muy sencillo | Documentación desorganizada |
| POO moderna | Hay que compilar las extensiones PHP |
| Muy rápido | |

Phalcon es un framework PHP implementado en C, lo que permite que se ejecute mas cerca del lenguaje lógico del sistema, llegando a conseguir una gran velocidad de ejecución. El problema viene cuando intentamos probarlo en un entorno de producción. Debido a su desarrollo es necesario compilar sus extensiones y si hay algún problema se tiene que debugar en C. Esto hace que su instalación sea bastante compleja. Además, hace imposible de instalar en un servidor compartido cosa que inhabilita completamente esta opción debido a que el servidor que nos proporciona la UB para el proyecto es un servidor compartido.

5.3. ¿Porque Laravel?

Como hemos podido observar en el apartado anterior al analizar los diferentes frameworks, observamos que nos encontramos con varias posibilidades a la hora de elegir un framework de desarrollo pero de momento ninguno ha acabado de encajar con nuestro objetivo. Laravel sin embargo pinta de otra manera.

| Ventajas | Inconvenientes |
|--|-----------------------|
| Framework moderno | Framework joven |
| Composer | ORM propio |
| POO moderna | Plantillas propias |
| Excelente documentación | |
| Gran comunidad de soporte | |
| Patrones de diseño elegantes | |
| Curva de aprendizaje moderada | |
| Módulos externos fácilmente acoplables | |

Laravel al considerarse un framework joven, puede presentar algunas carencias, pero debido al continuo desarrollo del proyecto Laravel (versión estable cada 6 meses) estas se han ido supliendo a lo largo de sus versiones estables. Una de las grandes ventajas de utilizar este framework para el desarrollo de aplicaciones es la gran comunidad que lo soporta. Gracias a esto se puede contar con una excelente documentación de respaldo y unos foros (Laracast) dedicados a ayudar a los usuarios. La curva de aprendizaje es moderada y anima a seguir trabajando con el framework a medida que se van descubriendo como implementar partes de la aplicación que se esta construyendo gracias a sus patrones de diseño elegantes y sencillos que no hacen mas que ayudar a entender el funcionamiento interno de la aplicación.

La capacidad de acoplar y utilizar módulos externos es increíblemente útil. Laravel utiliza componentes del paquete de Symfony [32], Doctrine y Monolog entre otros. Aprovecha estos componentes ya creados para no tener que implementarlos de nuevo y así poder centrarse en proporcionar la experiencia de usuario deseada en la cual se basa su filosofía.

En los siguientes apartados se profundizará en las características que hacen posible que Laravel destaque de esta manera como uno de los mejores Frameworks PHP para el

desarrollo de aplicaciones.

5.4. Patrones del desarrollo web y buenas practicas en Laravel

Laravel abraza una filosofía general que da prioridad a la creación de un código mantenible durante un largo periodo. Siguiendo las guías que se ofrece, el desarrollador debería ser capaz de crear código y sentirse libre de cambiarlo sin tener miedo de estropear la funcionalidad implementada. Para conseguir esto Laravel adopta una serie de patrones de desarrollo y unas buenas prácticas que vemos a continuación.

5.4.1. “Single responsibility patern”

Laravel esta basado en la arquitectura Modelo – Vista – Controlador (MVC). Esta es una arquitectura muy buena para el punto de vista del desarrollador porque separa cada componente de la web en un código base que es gestionado por separado sin temor al mal funcionamiento de las otras partes de la plataforma.

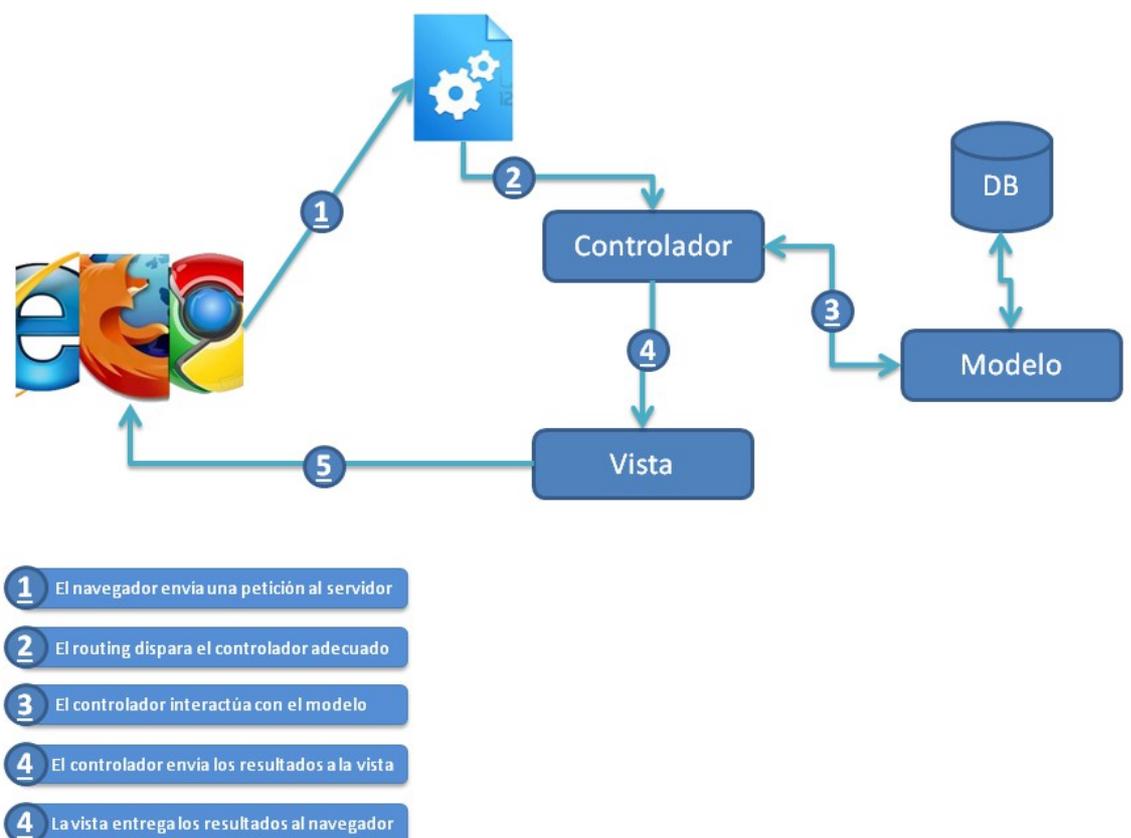


Figura 16. Arquitectura MVC de Laravel

5.4.2. “Convention over configuration”

Laravel define la estructura del directorio de nuestra aplicación por nosotros y introduce una serie de convenciones para el nombre de archivos, clases y tablas de base de datos. De esta manera toma ventaja de estas convenciones para atar y conjuntar la aplicación sin que sea necesaria una mayor configuración manual.

Java, Python u otros Frameworks PHP normalmente requieren muchos archivos XML de configuración. Laravel, sin embargo, no requiere ninguno.

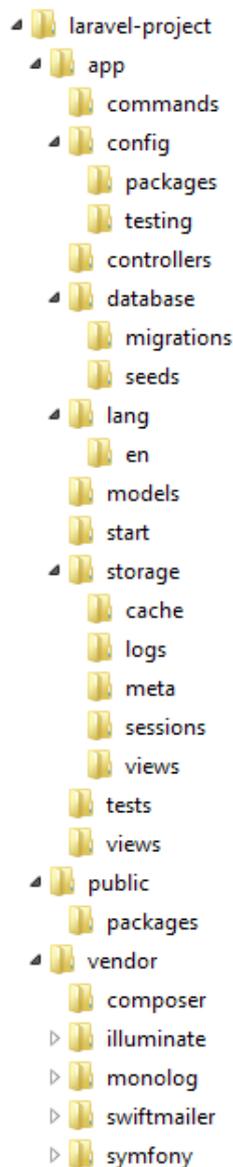


Figura 17. Estructura de un proyecto Laravel

Puede dar la sensación al principio de que Laravel organiza nuestro proyecto de una manera que no deseamos, pero todo lo contrario. Rápidamente nos damos cuenta que

esta organización hace que la velocidad de desarrollo de la aplicación aumente de una manera consistente y crea un código base mucho mas consistente para futuros trabajos en equipo o otros proyectos. Eligiendo “*smart defaults*” Laravel nos permite centrarnos en las funcionalidades de la aplicación mediante un código limpio y fácil de mantener.

5.4.3. “Don't repeat yourself”

El principio DRY (don't repeat yourself) consiste en escribir código limpio una vez y sólo una vez. Laravel promueve este principio proveyendo un entorno de programación que permite consolidar código compartido entre diferentes partes de la aplicación.

5.5. Composer & Laravel

Laravel depende de un cierto número de paquetes externos para incrementar su funcionalidad. Tal y como se ha comentado en el apartado 4, se utiliza Composer como gestor de dependencias. Esto hace que sea muy sencillo crear un proyecto y tenerlo listo para empezar a desarrollar en cuestión de minutos. Ya no será necesario descargar un archivo comprimido desde la web o clonar el repositorio de Github del paquete que necesitemos, con composer todo lo que tenemos que hacer es lo siguiente:

```
MacBook-Pro-de-Javi:~ javisanchez$ php composer.phar create-project laravel/laravel your-project-name
--prefer-dist
Installing laravel/laravel (v5.0.22)
- Installing laravel/laravel (v5.0.22)
  Downloading: 100%

Created project in your-project-name
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
- Installing jakub-onderka/php-console-color (0.1)
  Downloading: 100%

- Installing vlucas/phpdotenv (v1.1.0)
  Downloading: 100%

- Installing symfony/var-dumper (v2.6.4)
```

Figura 18. Creación de un nuevo proyecto mediante Composer

Esto nos dará una copia completa de Laravel, creando un nuevo proyecto. Seguidamente tan solo tendremos que ejecutar un par de instrucciones mas para tener nuestro proyecto preparado.

```
MacBook-Pro-de-Javi:your-project-name javisanchez$ php composer.phar install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Nothing to install or update
Generating autoload files
Generating optimized class loader
MacBook-Pro-de-Javi:your-project-name javisanchez$ php composer.phar update
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Removing vlucas/phpdotenv (v1.1.0)
- Installing vlucas/phpdotenv (v1.1.1)
  Downloading: 100%

- Removing symfony/var-dumper (v2.6.4)
- Installing symfony/var-dumper (v2.6.9)
  Downloading: 100%

- Removing symfony/translation (v2.6.4)
- Installing symfony/translation (v2.6.9)
  Downloading: 100%
```

Figura 19. Update de las dependencias

Composer install procesa el fichero *composer.json* y resuelve las dependencias del proyecto Laravel descarga, normalmente, en el directorio */vendor* todas las dependencias que requiere para funcionar. Finalmente Composer update actualiza las dependencias a la versión indicada. El núcleo de Laravel incluye paquetes externos y estos son acoplados mediante Composer durante la instalación. Entre estos paquetes encontramos:

- Symfony
- Monolog
- Doctrine

Aparte de la configuración inicial, si queremos añadir algún paquete adicional a nuestro proyecto, tan solo tenemos que editar el archivo *composer.json*, añadir la dependencia necesaria y volver a ejecutar el comando *composer update*.

```

{
  "name": "laravel/laravel",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",
  "type": "project",
  "require": {
    "laravel/framework": "4.1",
    "cartalyst/sentry": "2.1.*",
    "webpatser/laravel-uuid": "1.*",
    "fideloper/proxy": "~2.0"
  },
  "autoload": {
    "classmap": [
      "app/commands",
      "app/controllers",
      "app/controllers/Admin",
      "app/models",
      "app/database/migrations",
      "app/database/seeds",
      "app/tests/TestCase.php"
    ]
  }
}

```

Figura 20. Composer.json con diferentes paquetes

Adicionalmente todos los componentes que utiliza Laravel están disponibles en el repositorio Illuminate de Github [36].

5.6. Artisan

Artisan es el nombre de la línea de comandos incluida por Laravel. Proporciona un gran número de comandos para el programador mientras este está desarrollando su aplicación. Esta impulsado por el componente Console de Symfony y los métodos mas comunes se han dotado de una mayor fluidez para adaptarse al estilo de Laravel.

Ofrece comandos muy útiles que pueden ayudar a realizar diversas tareas, tales como generar migraciones o publicar recursos de un paquete. Adicionalmente a los comandos que vienen por defecto, el desarrollador puede incluir los suyos propios.

5.7. Migrations

5.7.1. ¿Que son las migraciones?

Cuando se trabaja con bases de datos durante el desarrollo de una aplicación esta tiene muchas interacciones, se crean nuevas tablas, se renombran columnas, se eliminan índices etc. En este área no podemos utilizar un sistema de control de

versiones como haríamos en otros apartados del proyecto, aunque hay ciertas ocasiones que necesitamos maneras sofisticadas de tener controlada la trayectoria y los cambios que sufren nuestras bases de datos:

- Cuando en un equipo de desarrolladores, cada persona tiene que saber cualquier cambio en el esquema de la base de datos.
- Cuando se hace un deploy en un servidor de producción y se necesita una manera robusta de hacer una actualización de la base de datos.
- Al trabajar en diversas máquinas, es necesario tener todas las bases de datos sincronizadas.

Poder realizar esta sincronización de bases de datos puede llegar a ser un gran problema, ya que si por ejemplo se realizan cambios en una base de datos y el código que la acompaña tiene que volver a una versión anterior, nos aparece un gran problema al tener un sistema descompensado en cuanto a versiones de código y base de datos.

Las migraciones son la manera que tiene Laravel de ayudarnos a evolucionar el esquema de nuestra aplicación web sin tener que eliminar y recrear la base de datos cada vez que se realiza un cambio. Esto significa no perder los datos de la base de datos, ya que, los únicos cambios que se realizan al ejecutar una migración son aquellos necesarios para mover el esquema de la base de datos de una versión a otra, tanto para adelante como para atrás en el tiempo. Se podría decir que Laravel consigue simular un sistema de control de versiones de las bases de datos.

Las migraciones son archivos que contienen una definición de clase con los métodos `up()` y `down()`.

El método `up()` se ejecuta cuando la migración se lanza para aplicar los cambios en la base de datos. El método `down()` se ejecuta para revertir los cambios. Si por ejemplo necesitamos actualizar nuestra base de datos, solamente tenemos que crear una nueva migración y ya lo tenemos listo, pudiendo actualizar a los nuevos datos y volviendo para la versión anterior cuando veamos necesario.

```
<?php

use Illuminate\Database\Migrations\Migration;

class AddTitleToUsers extends Migration {

    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('users', function($table)
        {
            $table->string('title');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('users', function($table)
        {
            $table->dropColumn('title');
        });
    }
}
```

Figura 21. Ejemplo de una migración

De esta manera Laravel nos permite cambiar el esquema de nuestra base de datos utilizando PHP en vez de SQL. Esto permite, gracias al Schema Builder [39] de Laravel, crear tablas en la base de datos, insertar columnas o índices de una manera muy rápida.

5.7.2. Crear y ejecutar migraciones

A la hora de crear migraciones, Laravel ha introducido una convención en cuanto al formato del nombre que deben de tener cada una de ellas. El nombre de una migración suele parecerse a la siguiente:

2015_06_01_161424_create_substance_table.php

Donde vemos que el formato del nombre es el timestamp del momento en que se crea la migración, seguido del nombre que se le otorga a esta:

`YYYY_MM_DD_HHMMSS_some_meaningful_name.php`

De esta manera se evitan colisiones y a la hora de ejecutar las migraciones se tiene una referencia de la antigüedad de cada una de ellas.

Para crear una migración Laravel nos permite hacerlo a través de su línea de comandos, Artisan.

Internamente Laravel utiliza una tabla especial para llevar el control de las migraciones creadas. Para crear esta tabla tan solo tenemos que ejecutar el comando:

```
$ php artisan migrate:install
Migration table created successfully.
```

Figura 22. Creación de la tabla migrations mediante Artisan

Crear una migración es tan sencillo como ejecutar el comando `migrate:make`

```
$ php artisan migrate:make create_users
Created Migration: 2013_06_30_124846_create_users
Generating optimized class loader
Compiling common classes
```

Figura 23. Creación de una migración

Una vez hemos creado las migraciones que queremos en nuestro sistema tenemos a nuestra disponibilidad los que son los 3 comandos que más utilizaremos durante el desarrollo de nuestra aplicación en cuanto al uso de las migraciones:

- ***php artisan migrate***

El comando `migrate` ejecuta las migraciones que hayamos creado en nuestro sistema y actualiza el esquema de nuestra base de datos con ellas.

- ***php artisan migrate:rollback***

El comando `rollback` tira para atrás la última migración ejecutada. Esto permite retroceder en el tiempo en caso de que haya surgido un error en nuestra última migración, permitiendo así deshacer el cambio efectuado y pudiendo ejecutar la migración correcta posteriormente. Para Laravel la última migración efectuada consiste en toda la operación de la migración, por lo que si en la acción que comentamos se han ejecutado 15 migraciones, se retrocederán las 15

migraciones.

- ***php artisan migrate:reset***

Reset retrocederá todas las migraciones que se hayan ejecutado (no solo las últimas) dejando así limpia la base de datos.

5.7.3. Seeds

Una vez tenemos la bases de datos con las tablas necesarias vemos que nos faltan los datos en estas. Laravel nos ofrece la clase Seeder para facilitarnos la tarea de llenar estas tablas de bases de datos. Esto es especialmente útil cuando por ejemplo tenemos una tabla de recursos y queremos añadir varios a la vez . La clase Seeder nos permite crear todos los registros que queramos haciendo uso de Fluent Query Builder y ejecutar la consulta creada.

```
class SubstanceSeeder extends Seeder {
    public function run()
    {
        DB::table('substances')->delete();

        Substance::create ( array(
            'cas' => '12002-61-8',
            'name' => 'actinium (III) oxide',
            'updated_at' => new DateTime,
            'created_at' => new DateTime,
        ));
        Substance::create ( array(
            'cas' => '27039-77-6',
            'name' => 'silver tetrachloroaluminate',
            'updated_at' => new DateTime,
            'created_at' => new DateTime,
        ));
        Substance::create ( array(
            'cas' => '7785-23-1',
            'name' => 'silver bromide',
            'updated_at' => new DateTime,
            'created_at' => new DateTime,
        ));
    }
}
```

Figura 24. Ejemplo de un fichero de Seed

Una vez creado el fichero seed lo ejecutamos mediante el comando

```
php artisan db:seed
```

De esta manera se añadirán todos los contenidos del fichero seed a las tablas de nuestra base de datos.

5.8. Eloquent

Eloquent es el ORM que proporciona Laravel. Un ORM o *Object Relation Mapping* es un modelo de programación que consiste en la transformación de las tablas de una base de datos en una serie de entidades que simplifican las tareas de acceso a los datos para el programador. Eloquent se posiciona en medio entre el desarrollador y la base de datos, transformando los datos de esta en entidades y permitiendo que el programador acceda a ellas como si fueran objetos.

Laravel llama a estas entidades Modelos y los utiliza para interactuar con su tabla correspondiente.

Obtener los datos para el programador es una tarea mucho mas sencilla utilizando Eloquent, en las siguientes imagen podemos comprobar la sencilla sintaxis de este:

```
$game = new Game;
$game->name = 'Assassins Creed';
$game->description = 'Assassins VS templars.';
$game->save();
```

```
$game = Game::find(1);
return $game->name;
```

```
$game = Game::find(1);
$game->delete();
```

Figura 25. Sintaxis Eloquent ORM

5.9. RESTful routing

Laravel hace uso del Restful routing [43] para gestionar sus rutas. Esto quiere decir que utiliza el conjunto de operaciones GET, POST, PUT y DELETE.

Las rutas definidas en las aplicaciones Laravel se encuentran en el archivo routes.php. Este, según la URI que capture, enviará a un controlador u otro para que se encargue de gestionar la petición de la manera correspondiente.

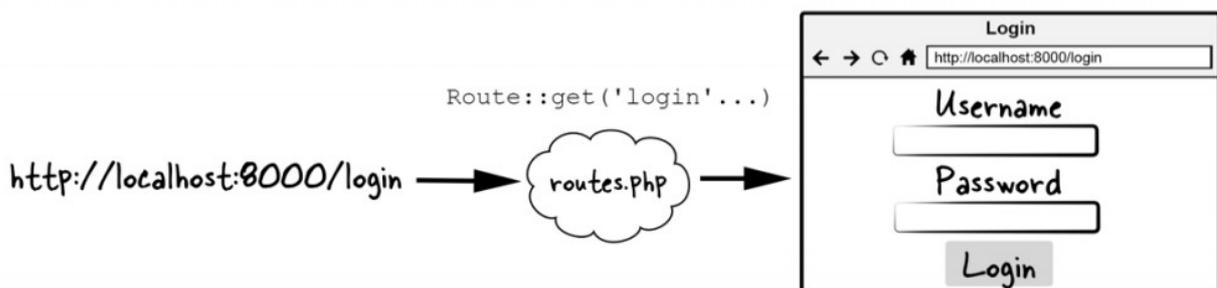


Figura 26. Arquitectura Routing Laravel

5.10. Blade Templates

Laravel incluye un motor de plantillas llamado Blade. Blade es básicamente un sub-lenguaje que antes de ser utilizado por la aplicación, es compilado a PHP. Gracias a el evitaremos el uso de tags php y podremos crear plantillas base e ir reutilizándolas a medida que las necesitemos. Estas plantillas son lo que se conoce como Layouts. Un Layout generalmente consiste en el header + footer que aparecen en todas las páginas de las aplicaciones. Blade nos permite crear plantillas “padre” y hacer que estas sean heredadas por otras plantillas, pudiendo así incluir su contenido en los “hijos”. De esta manera se le permite al programador hacer que una plantilla herede de varias, creando así diferentes secciones heredadas en la plantilla “hijo”.

Para el uso de Blade solo tenemos que crear nuestras plantillas con la extensión .blade.php en vez de .php y almacenarlas en el directorio views/.

```
@extends('admin._layouts.default')

@section('main')

    <div class="container">

        <div class="login_form">
            {{ Form::open(array('class'=>'form-horizontal well')) }}

            @if ($errors->has('login'))
                <div class="error_form">{{ $errors->first('login', ':message') }}</div>
            @endif

            <div class="control-group">
                {{ Form::label('email', 'Email', array('class'=>'control-label')) }}
                <div class="controls">
                    {{ Form::text('email', '', array('class' => 'form-control')) }}
                </div>
            </div>

            <div class="control-group">
                {{ Form::label('password', 'Password', array('class'=>'control-label')) }}
                <div class="controls">
                    {{ Form::password('password', array('class' => 'form-control')) }}
                </div>
            </div>

            <div class="form-actions">
                {{ Form::submit('Login', array('class' => 'btn btn-primary')) }}
            </div>

            {{ Form::close() }}
        </div>
    </div>

@stop
```

Figura 27. Ejemplo de plantilla Login.blade.php

5.11. Seguridad

5.11.1. Protección de rutas y autenticación de usuarios

El objetivo de la seguridad de Laravel es impedir que los usuarios que no tengan suficientes permisos no accedan a ciertas áreas de la aplicación.

Pueden utilizarse filtros en las rutas de la aplicación para permitir que sólo los usuarios autenticados o con suficientes permisos puedan acceder a la ruta indicada. Laravel proporciona el filtro auth por defecto, pero pueden crearse tantos filtros como se desee.

```
Route::get('profile', array('before' => 'auth.basic', function()
{
    // Sólo usuarios autenticados pueden ingresar
}));
```

Figura 28. Protección de una ruta mediante un filtro

```
Route::filter('auth.basic', function()
{
    return Auth::basic('username');
});
```

Figura 29. Definición del filtro de protección

En la aplicación Inventario de Química se ha creado un filtro para validar los permisos de cada usuario y así determinar si tienen acceso o no a la ruta concreta.

```
Route::filter('hasAccess', function($route, $request, $value)
{
    try
    {
        $user = Sentry::getUser();

        if (!$user->hasAccess($value))
        {
            return Redirect::to('access');
        }
    }

    catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
    {
        return Redirect::route('cms.login')->withErrors(array(Lang::get('user.notfound')));
    }
});
```

Figura 30. Filtro de acceso a las rutas según permisos de usuario

5.11.2. Seguridad de la aplicación

Por otra parte Laravel implementa internamente defensas ante ataques externos en la aplicación.

- Al utilizar Eloquent y Fluent Query Builder en los modelos de datos, protege la base de datos de ataques SQL injection escapeando cualquier tipo de input que pueda entrar por los formularios intentando insertar código SQL malicioso.
- Laravel utiliza una llave de autenticación, única por cada aplicación, para encriptar las cookies asegurándose de que no se puedan modificar externamente.
- También proporciona protección ante falsificaciones de peticiones (CSRF), asignaciones masivas y ataques por secuencia de comandos (XSS).

5.12. Laravel – El framework PHP mas famoso

Finalmente nos decantamos por Larvel. Como podemos observar en los siguientes gráficos, no somos los únicos, ya que en el 2015 Laravel se ha posicionado como el Framework mas popular y de referencia en el desarrollo de aplicaciones web.

PHP Framework Popularity at Work - SitePoint, 2015

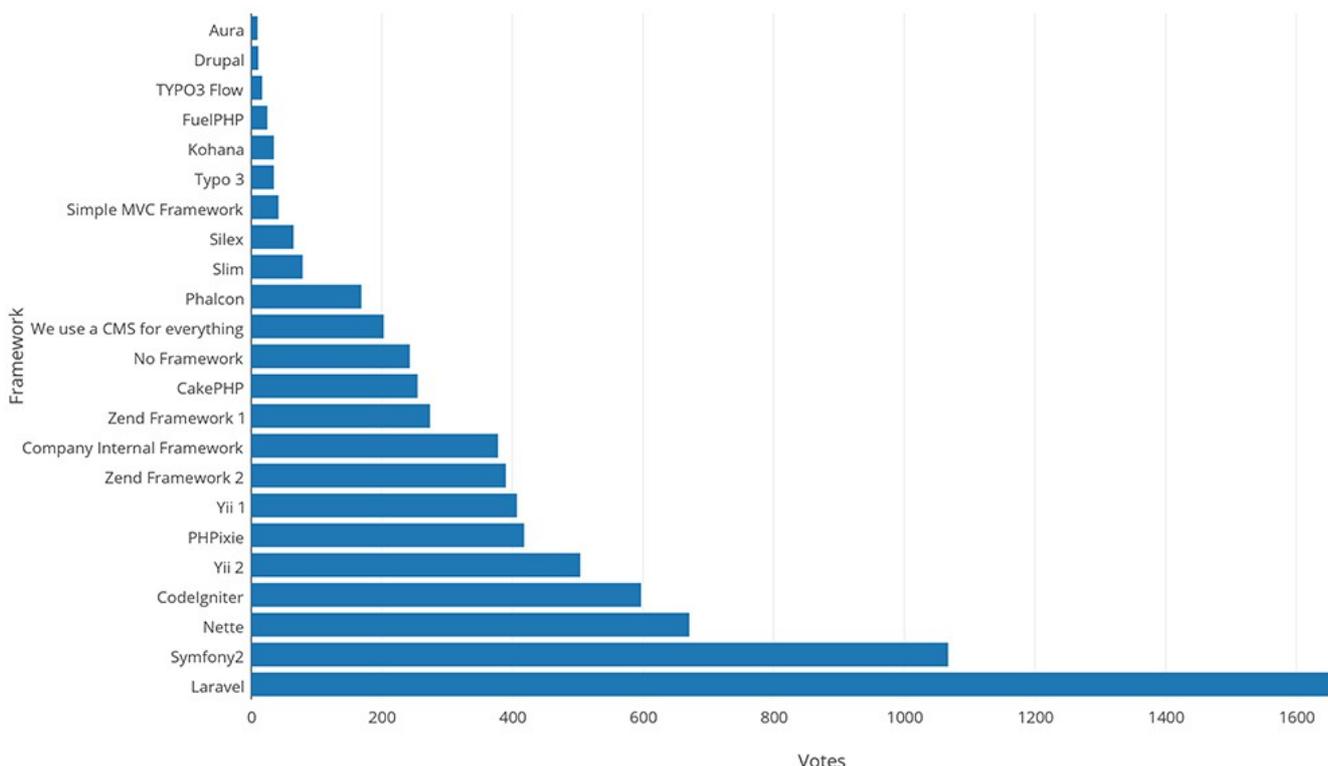


Figura 16. Popularidad Framework PHP en el trabajo

PHP Framework Popularity in Personal Projects - SitePoint, 2015

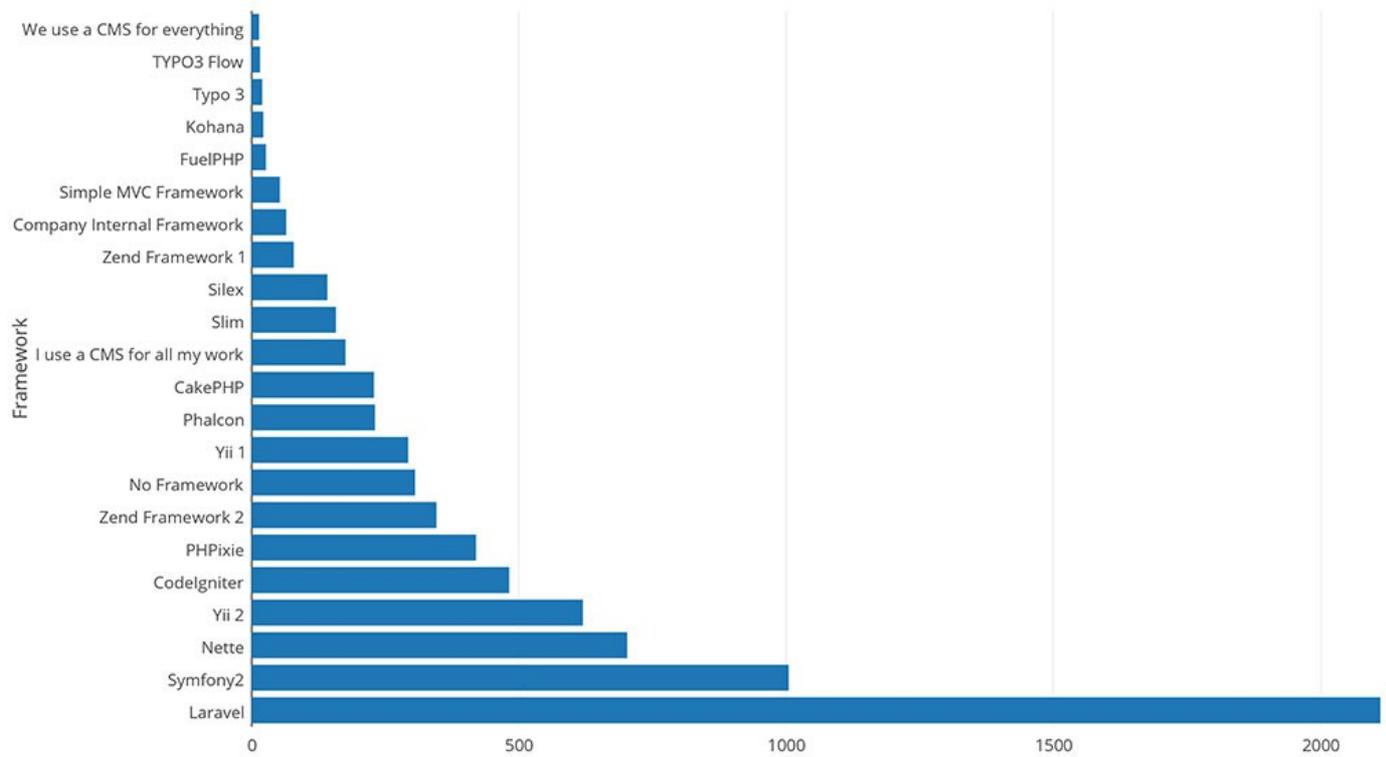


Figura 17. Popularidad Framework PHP en proyectos personales

6. Arquitectura

Tal y como se ha comentado en los apartados anteriores el proyecto Inventario de Química esta basado en la arquitectura del MVC.

6.1. Modelos

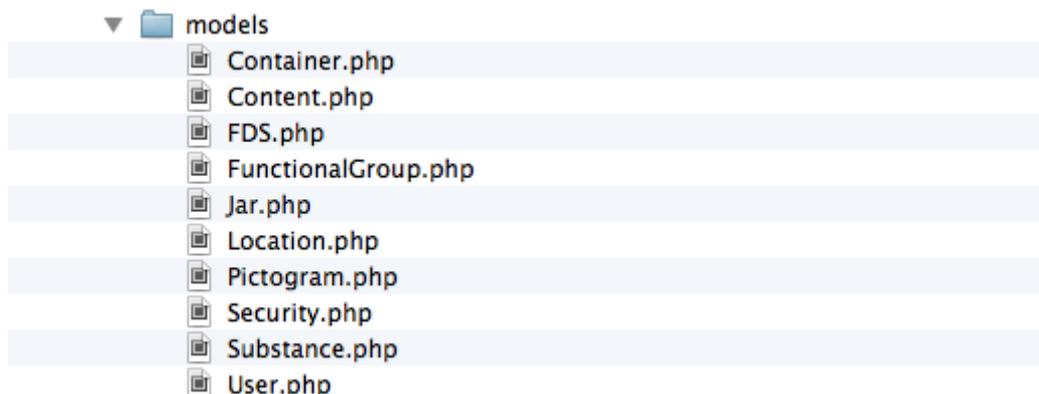


Figura 31. Modelos de la aplicación Inventario de Química

En el directorio models encontramos todos los modelos empleados. Cada uno de ellos corresponde a una entidad y cuentan con sus correspondientes atributos característicos.

6.2. Vistas

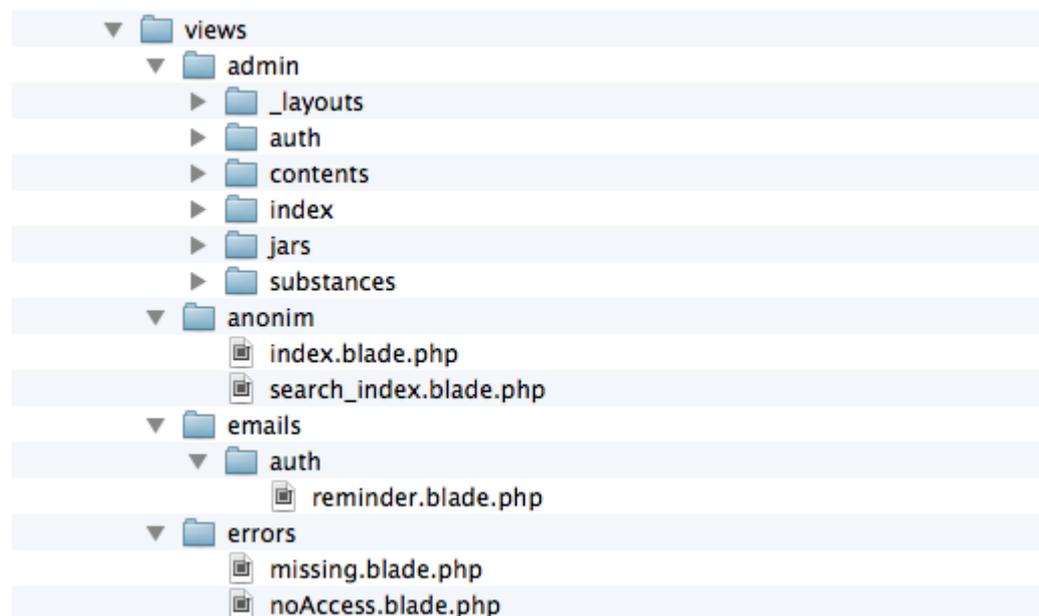


Figura 32. Vistas de la aplicación Inventario de Química

El directorio views contiene las diferentes plantillas que se utilizan en la aplicación. Estas heredan de la clase Blade y por lo tanto algunas de ellas heredan de otras, permitiendo así crear nuevas vistas que incluyen el contenido de otras.

6.3. Controladores

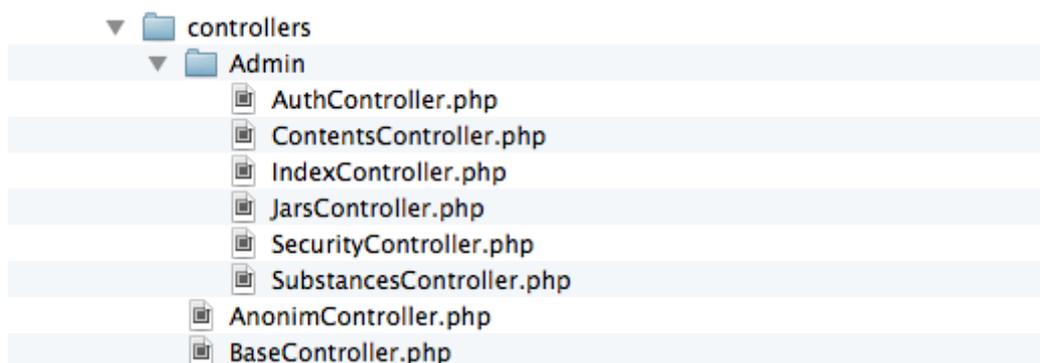


Figura 33. Diferentes controladores de la aplicación

Finalmente en el directorio de controllers encontramos los diferentes controladores de la aplicación. Se han estructurado en archivos separados según sus funciones para tener un mayor control sobre todos los modelos.

- *AuthController.php*: Gestión de la autenticación de los usuarios y la validación de estos.
- *ContentsController.php*: Gestión de todo lo relacionado con los contenidos (añadir, editar, eliminar etc).
- *IndexController.php*: Gestión de la ventana de administración principal de los técnicos de laboratorio y administradores.
- *JarsController.php*: Gestión de todo lo relacionado con los productos del sistema (añadir, editar, eliminar etc).
- *SecurityController.php*: En este controlador se han agrupado la gestión de los elementos de seguridad, así como localizaciones y contenedores.
- *SubstancesController.php*: Gestión de todo lo relacionado con las sustancias del sistema (añadir, editar, eliminar etc).
- *AnonimController.php*: Controlador del sistema al que pueden acceder los usuarios no autenticados.
- *BaseController.php*: Este es el controlador base del que heredan el resto de controladores.

7. Planificación

Para la planificación de este proyecto se dividió el proyecto en diferentes etapas:

- Etapa de **formación**: Aquí se incluye todo el trabajo de aprendizaje necesario para el proyecto.
- Etapa de **análisis** de requisitos: En esta etapa se incluye todo el trabajo de investigación realizado durante la duración del proyecto. Por lo tanto incluimos el análisis de los diferentes frameworks del mercado, la investigación sobre sustancias químicas, la seguridad asociada a estas etc.
- Etapa de **diseño** de la aplicación: Etapa en la cual se diseña la estructura de la base de datos y los diferentes modelos de la aplicación, así como el diseño del frontend y la arquitectura del proyecto.
- Etapa de **desarrollo**: En esta etapa contabilizamos el tiempo dedicado al desarrollo en sí de la plataforma y las diferentes pruebas realizadas.
- Etapa de **documentación**: Realización de la documentación necesaria para el proyecto, además de los manuales de usuarios de la plataforma.

En un primer momento se planteó la siguiente planificación para el proyecto, teniendo en cuenta estas 5 etapas:

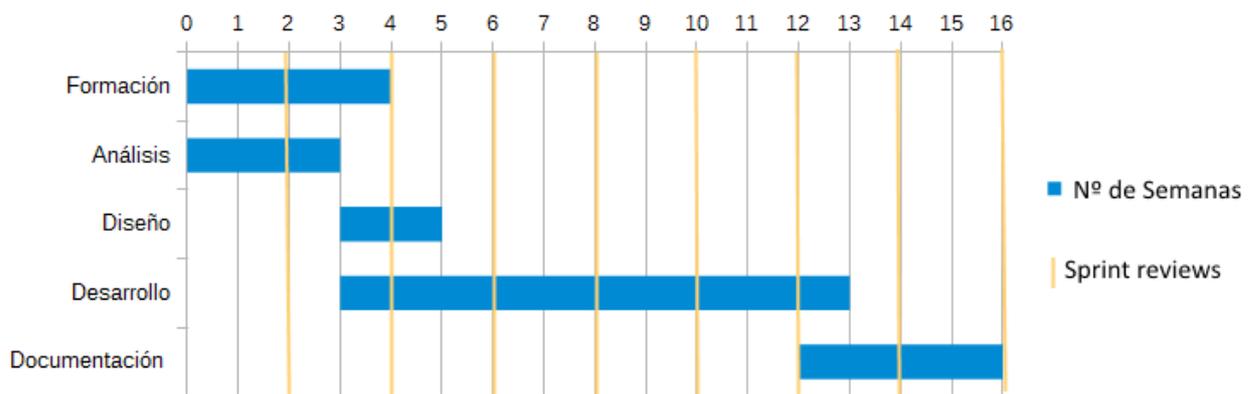


Figura 34. Diagrama de Gantt inicial

Finalmente se cumplieron las horas planteadas tanto en el diseño como en el desarrollo del proyecto. En las etapas de formación, análisis y documentación vemos que las horas son distintas:

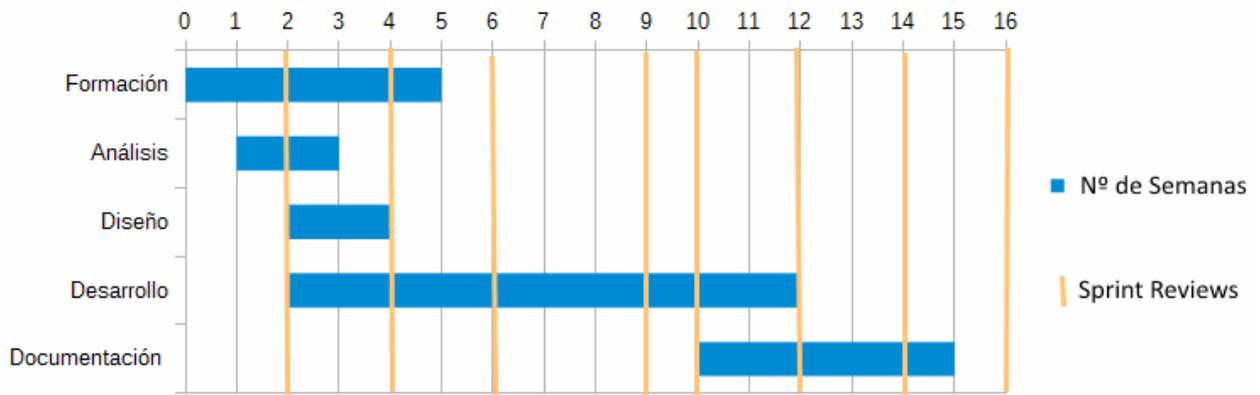


Figura 34. Diagrama de Gantt final

Esto se debe a que se a que durante la etapa de formación fueron necesarias mas horas de las previstas, ya que había diferentes elementos y ámbitos que se desconocían. Por otra parte la etapa de diseño no hizo falta que fuera tan extensa porque la elección del framework se solucionó más rápidamente que en la planificación inicial. Y por último la redacción de la memoria llevo mas tiempo del estimado inicialmente.

En cuanto a los sprint review, tal y como se aprecia en los diagramas, las reuniones programadas se tuvieron que ir adaptando según los horarios de los participantes teniendo en cuenta días festivos. En estas reuniones se verificaba el estado en que se encontraba la aplicación y si se habían cumplido los objetivos establecidos para esa reunión. Posteriormente se asignaban la continuación de los no cumplidos y se asignabas nuevas tareas. A excepción de una reunión, las demás se cumplieron tal y como se habían previsto en la planificación inicial.

8. Tests realizados

Durante el desarrollo del proyecto se han realizado una serie de test y de pruebas de uso para verificar el correcto funcionamiento y desarrollo de la aplicación.

Laravel incorpora herramientas de testing que nos permiten crear test de pruebas para verificar el resultado esperado en nuestra aplicación web.

Mediante estos test por ejemplo, podemos programar el resultado esperado al entrar en una ruta concreta y hacer que el test lo verifique por nosotros. De esta forma creando una batería de test podemos comprobar el correcto funcionamiento que esperamos al utilizar la aplicación.

Por otra parte, durante el desarrollo de la aplicación esta se ha testado con la herramienta de rendimiento web PageSpeed Insights [48]. Esta herramienta creada por Google analiza el contenido de una página web y a continuación da sugerencias para mejorar la velocidad de las páginas, tanto en ordenadores como en dispositivos móviles.

83 / 100 Resumen de sugerencias

! Elementos que debes corregir:

Evita los redireccionamientos a páginas de destino

› [Mostrar cómo corregirlo](#)

! Elementos que puedes plantearte corregir:

Especificar caché de navegador

› [Mostrar cómo corregirlo](#)

Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página

› [Mostrar cómo corregirlo](#)

Optimizar imágenes

› [Mostrar cómo corregirlo](#)

Habilitar compresión

› [Mostrar cómo corregirlo](#)

Minificar CSS

› [Mostrar cómo corregirlo](#)

Minificar HTML

› [Mostrar cómo corregirlo](#)

✓ 3 reglas aprobadas

Figura 35. Resultados PageSpeed Insights

Los resultados obtenidos indican que hay una redirección en nuestra aplicación web y por este motivo se nos penaliza. Esta redirección es necesaria debido al login de la UB, ya que al entrar a la dirección de la web, se nos redirecciona hasta él. Las demás sugerencias que se nos ofrecen son para comprimir las imágenes utilizadas, la compresión del CSS y HTML y el uso de un cacheado en el navegador. Estos puntos se comentarán en el apartado de mejoras de la aplicación mas adelante.

Finalmente, la plataforma ha sido testeada por algunos usuarios de la facultad de química para verificar el correcto funcionamiento de la aplicación y ver si había problemas de usabilidad durante su uso. Se pidió a los responsables del proyecto de química y a algunos profesores que hicieran pruebas de uso para verificar la aplicación, y por otra parte, se eligieron a becarios de la facultad de química que no tenían conocimiento alguno de la aplicación.

Los usuarios a los que se les ha pedido que realicen pruebas de uso en la aplicación pueden clasificarse en dos grupos.

- Usuarios que utilizarán la aplicación (a los que va dirigida la aplicación).
- Usuarios que no tienen conocimiento alguno de la aplicación (voluntarios a los que se les ha explicado en que consiste la aplicación y se le pide que ejecuten una serie de tareas mediante la plataforma).

Se han realizado los test con estas dos clases de usuarios para no condicionar las pruebas con previos conocimientos de la aplicación y descartar posibles favoritismos en cuanto a los requisitos.

A continuación veremos algunas de las correcciones mas destacadas que se que se realizaron gracias a los resultados de los test de los usuarios.

- **Grupos de usuarios**

Se detectó que al traspasar el proyecto al servidor de la UB e implementar el login de usuario de la UB los usuarios que pertenecían a un grupo de usuarios de la UB no podían acceder a la plataforma mediante su login. Esto se debía a que la versión de Apache2 instalada en el servidor de la UB no permite dar permisos a usuarios individuales y hacer un OR con grupos de usuarios. Para solucionar este problema, se implementaron listas de usuarios individuales y no por los grupos que proporcionaba la UB.

- **Buscador principal**

Las primeras veces que se utilizó el buscador principal se reportaron errores sobre el nombre de sustancias que mostraba sobre los productos que se buscaban, ya que buscaba el producto correctamente pero la sustancia que aparecía en ese producto no era la correcta. El problema era que al ejecutar la query de búsqueda el nombre de la primera sustancia sobre escribía al resto de sustancias y por este motivo no se mostraban.

- **Usabilidad en las frases de seguridad**

Los usuarios reportaron informes conforme era confuso para ellos que en las frases de seguridad que se pueden asignar a los producto hubiera un botón de eliminar inhabilitado. Este botón inicialmente se había inhabilitado para forzar que al menos hubiera una frase de seguridad por producto. Gracias a estos informes se verificó que esto causaba confusión entre los usuarios y por lo tanto se replanteó el diseño, pasando el peso de esta validación al controlador de productos para verificar que al menos hubiera una frase de seguridad por producto.

9. Conclusiones

Este proyecto buscaba la implementación de un sistema de gestión de productos, con una jerarquía determinada, para la Facultad de química que les permitiera administrar el inventario de productos químicos de la facultad mediante diferentes niveles de acceso en la aplicación.

Pese a que la aplicación se encuentra en sus primeras fases, ya que, mas adelante se ha planeado expandir e incrementar, consideramos que se han cumplido los objetivos propuestos durante el planteamiento del proyecto. Se han asentado unas firmes bases que permiten seguir añadiendo funcionalidades a medida de que la plataforma crezca y se expanda, además de permitir a la Facultad de química gestionar y administrar su inventario de productos tal y como se había planteado.

Por parte de la Facultad de Química comentar que se sienten muy satisfechos respecto al proyecto que se ha llevado a cabo ya que les permite satisfacer las necesidades y objetivos planteados inicialmente en el proyecto. Pese a que les hubiera gustado incluir mas funcionalidades que se han propuesto para una ampliación del proyecto por falta de tiempo, están muy ilusionados con poder aplicar este nuevo sistema de gestión a sus inventarios de productos y observar las mejoras que comporta su aplicación.

Como objetivos personales me había propuesto utilizar un framework de desarrollo web que permitiera la creación desde cero de esta plataforma adaptándola a un entorno de producción con un cliente real. Cabe destacar la importancia del proceso de selección del framework PHP, donde destacó Laravel y la profundización en él durante este proyecto. Personalmente considero que este objetivo se ha cumplido con creces, pese a que me hubiera gustado poder aplicar una serie de mejoras que no han podido ser abarcadas durante este periodo pero que se han planteado para su posterior desarrollo.

9.1. Mejoras propuestas y trabajo futuro

9.1.1. Mejoras en el Cacheado del sistema

Una de las mejoras a realizar en el sistema es la implementación del uso de la caché en Laravel, sobretodo para el uso de las consultas mas comunes realizadas en los buscadores de la aplicación, consiguiendo así una mayor rapidez en las búsquedas

mas comunes. También se tendrá que estudiar la posibilidad de cachear el contenido estático de la web mediante módulos externos de Laravel.

9.1.2. Mejora en el Registro de logs

Actualmente para ver el registro de los productos, cuando han sido modificados o cuando se ha creado un usuario se necesita acceder a base de datos y ver cuando ha sido modificado por última vez. Una mejora que se planteaba era la creación de logs periódicos donde la aplicación almacenaría información relevante que posteriormente se utilizaría internamente. Se había pensado en generar logs de acceso a la aplicación, así como acciones realizadas por los diferentes tipos de usuario. Al introducir el sistema de préstamos también se generarían logs con el historial de cada producto para poder consultar el recorrido de cada uno de ellos en caso necesario.

9.1.3. Mejoras en los buscadores

Debido a que las sustancias químicas se pueden nombrar de diferentes maneras y por diferentes sinónimos, se podría realizar una mejora en los buscadores que tuvieran en cuenta esto permitiendo al usuario buscar por diferentes nombres una misma sustancia que se encuentra en el inventario.

9.1.4. Creación y automatización de informes

Implementar un sistema de creación de informes internos permitiendo a usuarios determinados generar un informe con diferentes datos y estadísticas relacionadas con los productos del inventario. De esta manera se podrá visualizar de una manera mucho mas gráfica los productos que mas se han consumido, cuales se encuentran cerca de ser expirados, cuantas existencias quedan, el historial de un producto etc. Se generarían a petición del usuario y automáticamente cada ciertos periodos de tiempo.

9.1.5. Ampliación a diferentes laboratorios

Actualmente la plataforma solo se encuentra aplicada al los laboratorios de la facultad de química pero en fases posteriores se tiene planificado que incluya diversos

laboratorios externos, creando así una red interna entre ellos.

9.1.6. Sistema de préstamos

Al ampliar la plataforma a diferentes laboratorios se abre la posibilidad de la creación de un sistema de préstamos entre ellos. Este sistema consistiría en permitir a diferentes usuarios y laboratorios pedir prestados productos a otros laboratorios enviando una petición al sistema. Esta petición se aprobaría o no según un usuario gestor de préstamos y según el resultado de esta operación se cedería el producto en préstamo un tiempo determinado. Cuando este tiempo estuviera cerca de finalizar se notificaría al usuario de la aplicación de ello y este debería devolverlo.

10. Bibliografía

[1] “Capítulo 2: Inventario de Existencias”

<http://www.fao.org/docrep/w1604s/w1604s05.htm>

[2] “Mezclas y Soluciones Químicas”

<http://mezclasysolucionesquimicas.blogspot.com.es/>

[3] “Sustancia Química, Definición, Historia, Elementos Químicos”

http://docsetools.com/articulos-para-saber-mas/article_58646.html

[4] “Wikipedia: Mezclas y Sustancias Químicas”

<https://es.wikipedia.org/wiki/Mezcla>

https://es.wikipedia.org/wiki/Sustancia_qu%C3%ADmica

[5] “Sustancias simples y compuestas. Concepto | Química y algo mas”

<http://www.quimicayalgomas.com/quimica-general/sustancias-simples-y-compuestas-concepto/>

[6] “CAS Content”

<http://www.cas.org/content>

<http://www.cas.org/content/chemical-substances>

[7] “Common Chemistry – Search chemicals names and CAS Registry numbers”

<http://www.commonchemistry.org/>

[8] “Seguridad en el trabajo. Sustancias químicas”

<http://www.estrucplan.com.ar/Producciones/entrega.asp?IdEntrega=1360>

[9] “LAMP - ApacheMySQLPHP”

<https://help.ubuntu.com/community/ApacheMySQLPHP>

[10] “VMWare Virtualization for Desktop & Server, Application, Public & Hybrid Clouds”

<http://www.ub.edu/tecnicweb/category/caracteristiquesservei/>

[11] “Ubuntu 10.04.4 LTS (Lucid Lynx)”

<http://releases.ubuntu.com/10.04/>

[12] “Ubuntu 12.04.5 LTS (Precise Pangolin)”

<http://releases.ubuntu.com/12.04/>

[13] “Servei d'Hostatgeria web | Universitat de Barcelona”

<http://www.ub.edu/tecnicweb/category/caracteristiquesservei/>

[14] “Composer”

<https://getcomposer.org/doc/00-intro.md>

[15] “Packagist”

<https://packagist.org/>

[16] “Responsive web design”

http://www.w3schools.com/html/html_responsive.asp

[17] “Mobile First”

<http://zurb.com/word/mobile-first>

[18] “Get Bootstrap”

<http://getbootstrap.com/>

[19] “Bootstrap 3”

<http://www.esandra.com/disenio-web-responsive-usar-framework>

[20] “Ventajas de usar Bootstrap”

<http://jorgelessin.com/ventajas-y-desventajas-de-usar-bootstrap/>

[21] “Razones para desarrollar con Bootstrap”

<http://xn--diseocreativo-lkb.com/10-razones-para-desarrollar-una-web-con-bootstrap/>

[22] “GitHub About”

<https://github.com/about>

[23] “What exactly is GitHub”

<http://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>

[24] “Agile Manifesto”

<http://www.agilealliance.org/the-alliance/the-agile-manifesto/>

[25] “Que son las metodologias agiles”

<http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>

[26] “Introduction - Laravel The framework for web artisans”

<http://laravel.com/docs/4.1/introduction>

[27] “Laravel a beautiful PHP Framework”

<http://www.vladstudio.com/es/post/?laravel---a-beautiful-php-framework-that-does-not-make-me-feel-stupid>

[28] “Introduction to Laravel Framework”

<http://laravelbook.com/laravel-introduction/>

[29] “¿Que es Laravel?”

<https://laravelen espanol.wordpress.com/>

[30] “Symfony”

<https://symfony.com/>

[31] “Best php framework 2015 survey results”

<http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>

[32] “Laravel projects using symfony”

<http://symfony.com/projects/laravel>

[33] “From Cakephp to Laravel”

<http://www.slideshare.net/mccreaja/from-cakephp-to-laravel>

[34] “Comparing Laravel, Codeigniter, Cakephp”

<https://nerdmom.wordpress.com/2012/08/02/comparing-laravel-codeigniter-cakephp/>

[35] “What is phalcon PHP”

<http://brianglin.me/2014/01/what-is-phalcon-php/>

[36] “Laravel Components”

<https://github.com/illuminate>

[37] “Artisan Laravel Comands”

<http://code.tutsplus.com/tutorials/your-one-stop-guide-to-laravel-commands--net-30349>

[38] “PHP Master | Artisan and Laravel migrations”

<http://www.sitepoint.com/laravel-migrations/>

[39] “Laravel Schema Builder”

<http://laravel.com/docs/4.1/schema>

[40] “Code Bright for Laravel PHP”

<http://daylerees.com/codebright>

[41] “Laravel database seeding”

<http://laravelbook.com/laravel-database-seeding/>

[42] “Laravel Eloquent”

<http://laravel.com/docs/4.1/eloquent>

[43] “Representational State Transfer”

https://es.wikipedia.org/wiki/Representational_State_Transfer

[44] “Laravel first framework - routing”

<http://maxoffsky.com/code-blog/laravel-first-framework-chapter-3-routing/>

[45] “Plantillas blade en Laravel”

<http://uno-de-piera.com/plantillas-blade-en-laravel/>

[46] “Seguridad – Documentación Laravel”

<http://laraveles.com/docs/4.1/security>

[47] “Sistema de autenticacion usuarios laravel”

<https://styde.net/sistema-de-autenticacion-de-usuarios-en-laravel/>

[48] “PageSpeed Insights”

<https://developers.google.com/speed/pagespeed/insights/?hl=es>

11. Anexo

11.1. Proxy inverso servidor UB

Debido a que el servidor de producción es un servidor propio de la UB y ellos llevan la gestión de este servidor no se puede mostrar la configuración de este por motivos de seguridad. Sobre este servidor comentar que utiliza un proxy inverso y al realizar la subida a producción se tuvieron que realizar las siguientes configuraciones para que la aplicación funcionara correctamente.

El proxy inverso actúa como intermediario entre los clientes que quieren utilizar la aplicación web y los servidores web. Debido a esto cada petición que reciba el servidor web vendrá desde el intermediario en vez de hacerlo desde el cliente. Para solucionar los problemas que ocasionaba esta configuración se tuvo que recurrir a un paquete externo que incluía el siguiente servicio:

```
class ProxyServiceProvider extends ServiceProvider {  
  
    /**  
     * Indicates if loading of the provider is deferred.  
     *  
     * @var bool  
     */  
    protected $defer = false;  
  
    /**  
     * Register the service provider.  
     *  
     * @return void  
     */  
    public function register()  
    {  
        $request = $this->app['request'];  
        $proxies = $this->app['config']->get('proxy.proxies');  
  
        if( $proxies === '*' )  
        {  
            // Trust all proxies - proxy is whatever  
            // the current client IP address is  
            $proxies = array( $request->getClientIp() );  
        }  
  
        $request->setTrustedProxies( $proxies );  
    }  
}
```

Figura 36. ProxyServiceProvider

Para poder comprobar la dirección IP transmitida, Laravel necesita confiar en la dirección IP que le llega. Este paquete nos permitía definir una lista de “*trusted proxies*”, permitiendo así que Laravel la detecte como una IP de confianza, pudiendo buscar la dirección transmitida y configurarla como la verdadera dirección IP del cliente.

11.2. Configuración Apache

A continuación se muestra el archivo de configuración que se ha realizado en el servidor virtual en el que se ha desarrollado la aplicación.

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName inventari
  DocumentRoot /mnt/hgfs/inventari/public
  <Directory />
    Options FollowSymLinks
    AllowOverride All
  </Directory>
  <Directory /mnt/hgfs/inventari/public/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
  </Directory>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>

  ErrorLog /var/log/apache2/error.log

  # Possible values include: debug, info, notice, warn, error, crit,
  # alert, emerg.
  LogLevel warn

  CustomLog /var/log/apache2/access.log combined

  Alias /doc/ "/usr/share/doc/"
  <Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
  </Directory>
</VirtualHost>
```

Figura 37. vhosts apache

11.3. Manual de usuario

11.3.1. Introducción

La aplicación web Inventario de Química es un sistema de gestión de productos químicos guardados en los diferentes almacenes de la Facultad de Química. Los usuarios que tengan acceso a ella podrán consultar y modificar los productos almacenados en ellos.

11.3.2. Jerarquía de productos

Un producto es una unidad de almacenamiento de una o varias sustancias concretas en una localización específica y con unas características de seguridad asignadas. El sistema está organizado mediante la siguiente jerarquía de productos:

- Un producto está formado por un contenido y por un contenedor.
- Un contenido (especie) está formado por una o varias sustancias (cada una con su concentración determinada).
- Un contenedor es el envase del producto.

11.3.3. Acceso a la aplicación

El acceso a la aplicación se realizará mediante un explorador web, recomendable hacerlo con Firefox o Google Chrome. Una vez se haya accedido a la aplicación se mostrará la siguiente pantalla:

U
B Universitat de Barcelona

Català Español English

Identificació UB

Identifiqueu-vos amb el nom d'usuari i la contrasenya de la Intranet UB. Començareu una sessió que us donarà accés als serveis de la Universitat que requereixen identificació UB

Identificador

Contrasenya

Entra

Identificació alternativa

[Identificació STORK per a alumnes](#)

No podeu entrar?
[Consulteu l'ajuda](#)

© Universitat de Barcelona

Figura 38. Acceso UB

En esta pantalla se introducirá el usuario y contraseña de nuestro usuario de la UB. En el caso de profesores el usuario numérico y en el caso de alumnos el usuario que corresponde con 4 letras del abecedario.

11.3.4. Realizar una búsqueda

Una vez el usuario anónimo ha accedido en el sistema se le mostrará la página principal.

En ella podrá realizar las búsquedas sobre productos. Para ello bastará con elegir los criterios de búsqueda del producto (nombre de la sustancia, CAS, fórmula o hash del producto) y en el laboratorio donde quiera realizar la búsqueda. Finalmente quedará introducir la cadena de texto a buscar y realizar la búsqueda.

Al realizar la búsqueda se volverá a mostrar de nuevo la página principal, incluyendo esta vez los resultados de la consulta efectuada. En ella encontramos todas las opciones disponibles para este tipo de usuario, tales como realizar otra búsqueda, acceder a los detalles de un producto y acceder al login interno de la aplicación. Si se desea imprimir los resultados de la búsqueda efectuada, basta con imprimir página mediante el navegador y la aplicación dará el formato correspondiente a los resultados.

Search

Home | Log in

Facultat de Química

Login interno

return home

Results for: silver

Resultados de la búsqueda

Buscador

Detalles de producto

| Hash Code | Substance | CAS | Location | Container | State | Details |
|--------------------------------------|--|--|-----------------------------|--------------|---------|--------------------------|
| 8d33af9a-4c25-512d-9ea3-381a7e071fcb | • silver bromate - Concentration: 100 | CAS: 7783-89-3 | Magatzem de Productes - A23 | glass - 15ml | Active | See more |
| 189c3b8f-8a0c-516a-a592-c35044333922 | • silver bromate - Concentration: 100 | CAS: 7783-89-3 | Magatzem de Productes - A23 | glass - 15ml | Active | See more |
| ce5d7386-6e69-5998-92a5-a64fe1f30d07 | • silver bromide - Concentration: 30 • Substance test - Concentration: 35 • silver cyanide - Concentration: 35 | CAS: 7785-23-1 CAS: 123312 CAS: 506-64-9 | Magatzem de Productes - A09 | glass - 15ml | Retired | See more |

Figura 39. Búsqueda realizada por usuario anónimo

11.3.5. Login interno

En este apartado nos encontramos con el login interno de la aplicación. Solo tendrán usuario y contraseña aquellas personas que tengan un perfil de Técnicos de laboratorio o Administradores.

- **Técnicos de Laboratorio:** tendrán un rol de control y de mantenimiento de la aplicación, por lo que podrán añadir productos, copiar productos, añadir contenidos etc.
- **Administradores:** tendrán acceso completo a la aplicación. Son los encargados de corregir los posibles errores que no puedan solucionar los técnicos por temas de permisos. Es decir, se encargan de los apartados mas sensibles, tales como la seguridad.

11.3.6. Dashboard administración

En este apartado encontramos todas las opciones a las que pueden acceder los usuarios administradores. En este dashboard diferenciamos las tres pestañas que proporcionan información a los administradores sobre estado de algunos productos, también encontramos un buscador como en el usuario anónimo. Adicionalmente este

usuario tiene acceso a la creación de productos y a todas las demás opciones que aparecen en el desplegable en la barra de navegación.

The screenshot shows the 'Facultat de Química' dashboard. At the top left is the 'Universitat de Barcelona' logo. The top right navigation bar includes 'Home', 'More options', and 'Log out'. Below this is a 'Search' section with a text input field, a dropdown for 'Name of the substance', a dropdown for 'Magatzem de Productes', and 'Search' and 'Add Product' buttons. An arrow points to the 'More options' dropdown with the text 'Opciones disponibles'. Below the search section is a 'Products' section with tabs for 'Expired', 'About to Expire', and 'Lending'. An arrow points to the 'About to Expire' tab with the text 'Información sobre productos'. Below the tabs is a table of products. An arrow points to the 'Add Product' button with the text 'Añadir producto'.

| Hash Code | Substance | CAS | Location | Container | State | Details |
|--------------------------------------|---|-----------------|-----------------------------|-------------|--------|--------------------------|
| 9f9ab07a-5526-5ed7-91c6-82fc93d9674a | • actinium (III) oxide - Concentration: 5 | CAS: 12002-61-8 | Magatzem de Productes - A23 | glass - 1ml | Active | See more |
| 04bf830f-cc02-5085-913f-0215a9e69036 | • actinium (III) oxide - Concentration: 5 | CAS: 12002-61-8 | Magatzem de Productes - A23 | glass - 1ml | Active | See more |
| 18083ce5-937f-5972-8edb-72230f5ff7a0 | • actinium (III) oxide - Concentration: 5 | CAS: 12002-61-8 | Magatzem de Productes - A23 | glass - 1ml | Active | See more |

Figura 40. Dashboard administrador/Técnico

11.3.7. Añadir, Editar y Eliminar

Para añadir, editar o eliminar las entidades que comentaremos a continuación nos aparecerá un formulario muy parecido, por este motivo se agruparán en una imagen las tres opciones posibles para cada entidad.

11.3.7.1. Sustancias

En este apartado encontraremos un buscador exclusivo para sustancias y un listado donde aparecerán los resultados de las búsquedas realizadas. Para añadir una sustancia basta con apretar el botón verde Add Substance. Para editar y/o eliminar una sustancia hay que acceder a Edit, en el listado de sustancias que tenemos en este apartado.

Edit Substance

Eliminar sustancia  

Substance:



Name:

CAS:

Formula:

CDX File:

Observations:

Upload a picture

 No se ha seleccionado ningún archivo.

Upload a CDX File

 No se ha seleccionado ningún archivo.

Figura 41. Añadir/Editar/Eliminar sustancia

Como se puede observar en la anterior imagen se pueden editar o añadir los diferentes atributos que nos muestran, además de poder subir una imagen o un fichero CDX propio de la sustancia. Para guardar o editar sólo tendremos que dar a Save. Para eliminar bastará con apretar la imagen de la papelera.

11.3.7.2. Contenidos

Al igual que ocurre con las sustancias, en los contenidos encontramos un buscador especializado en contenidos. Mediante él podremos consultar Para añadir un contenido bastará con presionar el botón Add Content en el apartado de contenidos y este nos dirigirá a la vista correspondiente. Para eliminar o editar un contenido existente tendremos que presionar Edit, en el listado de contenidos que aparece en el apartado de contenidos.

Figura 42. Añadir/Editar/Eliminar contenido

Una vez nos encontremos en el formulario anterior, añadiremos o modificaremos los datos que nos interesen. Para agregar mas de una sustancia al contenido actual bastará con presionar el botón + y nos aparecerá una nueva. Tendremos que seleccionar la sustancia correspondiente e indicar el porcentaje de concentración que presentará en este contenido. Finalmente guardaremos la sustancia en el sistema. Si por otra parte queremos eliminarla, bastará con presionar el icono de la papelera.

11.3.7.3. Localizaciones y contenedores

Se han agrupado las localizaciones y los contenedores debido a que sus formularios de agregación, edición y eliminación son muy similares.

Para añadir una de estas entidades sólo tendremos que acceder a la página correspondiente de la entidad y ahí nos aparecerá el formulario. Si en cambio queremos editar o eliminar deberemos darle a Edit en el listado de entidades que se nos muestra en la parte inferior de la página correspondiente a la entidad a que nos estamos refiriendo.

Figura 43. Añadir/Editar/Eliminar Localización

Facultat de Química
Home | More options | Log out

Universitat de Barcelona

Edit Container Eliminar

Atributos

Name
container 4

Material
plastic

Capacity [Unit of volume]
100g

Properties

Save Cancel

Figura 44. Añadir/Editar/Eliminar Contenedor

11.3.7.4. Seguridad, pictogramas y características

Al igual que ocurre con los contenedores y las localizaciones, toda la seguridad la agruparemos en este apartado.

Al ir a la página de una de las entidades comentadas anteriormente vemos un formulario para añadir más entidades y en la parte inferior un listado de las entidades. Para añadir otra sólo tendremos que introducir los datos y enviar el formulario. Si por otra parte queremos editar o eliminar deberemos apretar Edit y se nos mostrará uno de los siguientes formularios para realizar la acción deseada.

Facultat de Química
Home | More options | Log out

Universitat de Barcelona

Edit Pictogram Eliminar

Pictogram:

Cambiar imagen

Title
Corrosiu

Observations

Replace the Pictogram
Seleccionar archivo Ningún archivo seleccionado

Save Cancel

Figura 45. Añadir/Editar/Eliminar Pictograma

The screenshot shows the 'Edit Security' form within the 'Facultat de Química' web interface. The header includes the 'Universitat de Barcelona' logo and navigation links for 'Home', 'More options', and 'Log out'. The form contains a 'Type' dropdown menu set to 'P', a 'Code' text input field containing 'P314', and a 'Text' area with the content 'Consultar a un médico en caso de malestar.'. At the bottom, there are 'Save' and 'Cancel' buttons.

Figura 46. Añadir/Editar/Eliminar Frases de seguridad

The screenshot shows the 'Edit Characteristics' form. The header is identical to the previous figure. The form has a 'Name' text input field containing 'Àcid carboxílic' and an empty 'Observations' text input field. 'Save' and 'Cancel' buttons are located at the bottom of the form.

Figura 47. Añadir/Editar/Eliminar Características

11.3.7.5. Productos

Al igual que en las otras entidades, en los productos también contamos con un buscador interno que nos permite buscar productos por alguna de sus características. Para añadir un producto basta con apretar Add Product en la página principal y este nos conducirá a la siguiente vista:

Add Product

[return to Products](#)

| | |
|---|--|
| Content More info | Container More info |
| <input type="text" value="content 1"/> | <input type="text" value="container 1 - 1ml"/> |
| Manufacturer | Location More info |
| <input type="text"/> | <input type="text" value="Magatzem de Productes - A23"/> |
| Security | |
| <input type="checkbox"/> Add security to the Product | |
| <input type="text" value="R1 - Explosive when dry"/> <input type="button" value="X"/> | |
| Pictograms | |
| <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  | |
| Add FDS file | Advertise Word |
| <input type="button" value="Examinar..."/> No se ha seleccionado ningún archivo. | <input type="text" value="-"/> |
| Expire Date | |
| <input type="text" value="Pick the date"/> | |
| State | Use |
| <input type="text" value="Active"/> | <input type="text" value="Sealed"/> |
| <input type="button" value="Add Product"/> | <input type="button" value="Cancel"/> |

Figura 48. Añadir Producto

Para crear un producto tendremos que elegir mediante los selectores que nos muestran los diferentes contenidos, contenedores y localizaciones. Para ver mas información sobre estos se puede accionar More info. Una vez elegidos, pasamos a la seguridad. En este apartado podemos seleccionar tantas frases de seguridad como deseemos, pudiendo eliminar o añadir mas según nuestro criterio. Finalmente añadimos el resto de datos del producto y lo guardamos en el sistema.

Para ver el detalle de un producto y así acceder a su información, además de la posibilidad de editarlo, copiarlo y/o eliminarlo, solo tenemos que darle a See more en un listado de productos del producto que nos interese, de la misma manera que lo haría un usuario anónimo.

Details Editar, Copiar y Eliminar producto   

[return to products](#)

Todo sobre el producto 

Generador de códigos QR  [Generate QR](#)

Product details

Hash Code:

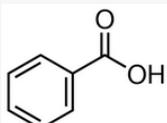
State: Use:

Manufacturer: Expire date: Retire date:

Content of the Product

Name: State:

Substances in the Content

 CAS: Name: Concentration [%]:

CDX File: [benzoic acid.pdf](#) Formula: Characteristics: Àcid carboxílic

Figura 49. Detalles del Producto

En los detalles del producto aparecerán todos los atributos de este, así como las diferentes opciones de Editar, Copiar y Eliminar el producto que estamos visualizando. Al editar un producto se nos mostrarán los campos de este y una vez introducidos los valores deseados podremos guardarlo. Al copiar el sistema creará un producto idéntico al seleccionado y nos permitirá editar el nuevo producto con los datos que se deseen. Finalmente para eliminar un producto bastará con accionar el botón con forma de papelera.

La generación de códigos QR se comenta en el siguiente apartado.

11.3.7.6. Códigos QR

La aplicación web permite la generación de códigos QR que al generarse permite el acceso a los detalles de un producto concreto. La idea es generar estos códigos QR y engancharlos físicamente a los productos químicos para posteriormente, en el almacén, poder reconocerlos mediante una aplicación desde un dispositivo móvil. Para reconocer un producto con un código QR, se tendrá que tener el producto con el código y un dispositivo móvil con cámara o un lector de códigos QR conectado a un ordenador. Se escanea con una aplicación especializada en la lectura de códigos QR (*Barcode Scanner*) o con el lector y seguidamente nos aparecerá el link para acceder a los

detalles del producto escaneado desde nuestro dispositivo móvil o desde el ordenador. Para generar el código QR basta con ir a los detalles del producto deseado y pulsar el botón Generate QR. Una vez se nos muestra el código podemos guardarlo como una imagen o imprimirlo directamente mediante el explorador web que utilicemos.

11.4. Casos de uso textuales

11.4.1.1. Login UB

Actores: Usuario no conectado

Descripción: Usuario sin identificar podrá conectarse a la aplicación con un usuario y una contraseña.

Pre-condición: -

Flujo Básico:

1. El usuario introduce los datos para acceder en un formulario de log in de la UB.
2. Si los datos introducidos son correctos, la aplicación le redirigirá a la aplicación.

Flujo Alternativo:

2. Si los datos introducidos no son correctos, la aplicación marcará un acceso denegado mostrando un error de permisos para acceder.

Pos-condición: El usuario sin identificar pasa a convertirse en un usuario profesor.

11.4.1.2. Login aplicación

Actores: Usuario profesor.

Descripción: Un usuario identificado que quiere acceder al apartado de Técnico de Laboratorio o Administrador del sistema

Pre-condición: Haber accedido mediante el login de la UB.

Flujo Básico:

1. El usuario introduce los datos en el formulario de log in de la aplicación.
2. Si los datos introducidos son correctos la aplicación redirigirá al usuario a la ruta correspondiente según su rol asignado.

Flujo Alternativo:

2. Si los datos introducidos son incorrectos la aplicación volverá al formulario de login y mostrará que datos son erróneos.

Pos-condición: El usuario profesor pasa a convertirse en usuario Técnico o Administrador.

11.4.1.3. Realizar búsquedas en el inventario

Actores: Usuario profesor, usuario Técnico y usuario Administrador.

Descripción: Un usuario identificado en la aplicación quiere realizar una búsqueda en el inventario por uno de los posibles criterios de búsqueda permitidos (nombre de la sustancia, CAS, Formula, código HASH y localización del producto).

Pre-condición: Haber accedido mediante el login de la UB y encontrarse en la página índice de la aplicación.

Flujo Básico:

1. El usuario elige un criterio de búsqueda (puede combinar cualquier criterio con la localización), introduce la cadena de texto que desea y pulsa el botón "Search" para

realizar la búsqueda.

2. Si la búsqueda da resultados, se mostrarán en la pantalla del usuario los productos que concuerdan con la búsqueda realizada. En esta pantalla de resultados se seguirá mostrando el formulario de búsqueda y la búsqueda anterior realizada, por si se quiere realizar otra consulta al visualizar los resultados de la anterior sin tener que volver atrás.

Flujo Alternativo:

2. Si no hay ningún resultado de búsqueda que concuerde con los parámetros introducidos no se mostrará ningún resultado pero se seguirán mostrando los datos por los que se han buscado.

Pos-condición: El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.

11.4.1.4. Imprimir resultados de búsqueda

Actores: Usuario profesor, usuario Técnico y usuario Administrador.

Descripción: Un usuario identificado en la aplicación que ha realizado una búsqueda quiere imprimir los resultados de esta.

Pre-condición: Haber accedido mediante el login de la UB y haber realizado una búsqueda en la aplicación.

Flujo Básico:

1. El usuario imprime los resultados de búsqueda y elige en que tipo de formato visualizar estos.

Flujo Alternativo: -

Pos-condición: El usuario ha guardado o imprimido los resultados de su búsqueda.

11.4.1.5. Visualizar detalles de Producto

Actores: Usuario profesor, usuario Técnico y usuario Administrador.

Descripción: Un usuario identificado en la aplicación quiere visualizar los detalles de un producto concreto.

Pre-condición: Haber accedido mediante el login de la UB y haber realizado una búsqueda siendo usuario profesor o visualizar uno de los listados disponibles en los usuarios técnico o administrador.

Flujo Básico:

1. El usuario elige la opción “See more” en los listados.
2. La aplicación cargará todos los detalles y características de ese producto en una nueva vista disponible para el usuario.
3. El usuario puede accionar el botón “return to products” para volver a la vista donde se encontraba anteriormente.

Flujo Alternativo:

2. Si el producto no existe, se le mostrará una página de error al usuario permitiéndole la opción de volver al inicio de la aplicación.

Pos-condición: El usuario ha visualizado los detalles del producto y puede volver a consultar los de cualquier otro a su disposición.

11.4.1.6. Generar código QR de un producto

Actores: Usuario profesor, usuario Técnico y usuario Administrador.

Descripción: Un usuario identificado en la aplicación quiere generar un código QR para identificar un producto concreto.

Pre-condición: Haber accedido mediante el login de la UB y encontrarse en la página de detalles de un producto.

Flujo Básico:

1. El usuario acciona el botón "Generate QR"
2. La aplicación generará un código QR y se lo mostrará al usuario en una pestaña nueva en el navegador.
3. El usuario podrá guardar el código QR como una imagen e imprimirlo.

Flujo Alternativo:

2. Si se produce algún error se mostrará al usuario una página informándole de ello.

Pos-condición: El usuario ha generado el código QR del producto deseado y puede utilizarlo.

11.4.1.7. Ver listado de productos Expirados, a punto de expirar y en préstamo.

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere consultar el listado de productos expirados, a punto de expirar y en préstamo.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Inicio de Administración.

Flujo Básico:

1. El usuario selecciona la pestaña (tab) que desee.
2. La aplicación mostrará al usuario un listado con los últimos productos que coincidan con el criterio de búsqueda de la pestaña seleccionada.

3. Al cambiar de pestaña se le mostrarán los productos característicos de la pestaña seleccionada.

Flujo Alternativo:

2. En caso de que no haya ningún producto en la pestaña seleccionada no se mostrará ningún producto.

Pos-condición: El usuario ha podido visualizar los productos según el criterio aplicado en cada tab.

11.4.1.8. Añadir Producto

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere añadir un producto al inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Producto o en la página de inicio de Administración.

Flujo Básico:

1. El usuario acciona el botón "Add Product".
2. El sistema mostrará al usuario la vista con el formulario con los datos necesarios a introducir.
3. El usuario introduce los datos del producto y envía el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el producto al sistema y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de productos.

4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido un nuevo producto al sistema.

11.4.1.9. Editar Producto

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere editar un producto existente del inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de detalle del producto.

Flujo Básico:

1. El usuario acciona el icono del lápiz en el detalle del producto.

2. El sistema mostrará al usuario la vista con el formulario de edición con los datos del producto seleccionado.

3. El usuario cambia los datos que considere oportuno y envía el formulario.

4. Si la validación de los datos introducidos es correcta, se añade el producto al sistema con los nuevos datos y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de productos.

4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al

punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha editado el producto seleccionado.

11.4.1.10. Duplicar Producto

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere duplicar y editar un producto del inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de detalle de Producto.

Flujo Básico:

1. El usuario acciona el icono de duplicar producto.
2. El sistema mostrará al usuario la vista del formulario de edición con las mismas características del producto copiado.
3. El usuario edita y cambia los datos del nuevo producto y envía el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el producto al sistema y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido un nuevo producto al sistema a partir de la copia de uno anterior.

11.4.1.11. Eliminar Producto

Actores: Usuario Administrador.

Descripción: Usuario de administración quiere eliminar un producto del inventario.

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página de detalle de Producto.

Flujo Básico:

1. El usuario acciona el icono de la papelera para eliminar el producto.

2. El sistema eliminará el producto seleccionado.

Flujo Alternativo: -

Pos-condición: El usuario ha eliminado un producto del sistema.

11.4.1.12. Realizar búsqueda interna de Productos

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere realizar una búsqueda por las características de un producto.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Productos.

Flujo Básico:

1. El usuario elige un criterio de búsqueda (Hash, Fabricante o Estado) introduce la cadena de texto que desea y pulsa el botón "Search" para realizar la búsqueda.

2. Si la búsqueda da resultados, se mostrarán en la pantalla del usuario los productos que concuerdan con la búsqueda realizada. En esta pantalla de resultados se seguirá mostrando el formulario de búsqueda y la búsqueda anterior realizada, por si se quiere realizar otra búsqueda al visualizar los resultados de la anterior sin tener que volver atrás.

Flujo Alternativo:

2. Si no hay ningún resultado de búsqueda que concuerde con los parámetros introducidos no se mostrará ningún resultado pero se seguirán mostrando los datos por los que se han buscado.

Pos-condición: El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.

11.4.1.13. Añadir Sustancia

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere añadir una sustancia al inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Sustancias.

Flujo Básico:

1. El usuario acciona el botón "Add Substance".
2. El sistema mostrará al usuario la vista con el formulario con los datos necesarios a introducir.
3. El usuario introduce los datos de la sustancia y envía el formulario.
4. Si la validación de los datos introducidos es correcta, se añade la sustancia al

sistema y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de sustancias.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido una nueva sustancia al sistema.

11.4.1.14. Editar Sustancia

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere editar una sustancia existente en el sistema

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de detalle del sustancia frente a un listado de sustancias.

Flujo Básico:

1. El usuario apretará "Edit" en la sustancia del listado que quiera editar.
2. El sistema mostrará al usuario la vista con el formulario de edición con los datos de la sustancia seleccionada.
3. El usuario cambiará los datos que considere oportuno y enviará el formulario.
4. Si la validación de los datos introducidos es correcta, se añade la sustancia al sistema con los nuevos datos y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de sustancias.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha editado la sustancia seleccionada.

11.4.1.15. Eliminar Sustancia

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere eliminar una sustancia existente en el sistema

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página de sustancias frente a un listado de sustancias.

Flujo Básico:

1. El usuario apretará "Edit" en la sustancia del listado que quiera eliminar.
2. El sistema mostrará al usuario la vista con el formulario de edición con los datos de la sustancia seleccionada.
3. El usuario seleccionara el icono de la papelera.
4. La sustancia seleccionada se eliminará del sistema.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de sustancias.

Pos-condición: El usuario ha eliminado la sustancia seleccionada.

11.4.1.16. Realizar búsqueda interna de Sustancias

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere realizar una búsqueda por las características de una sustancia.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Sustancias.

Flujo Básico:

1. El usuario elige un criterio de búsqueda (Nombre, Formula, CAS o Características) introduce la cadena de texto que desea y pulsa el botón "Search Substance" para realizar la búsqueda.
2. Si la búsqueda da resultados, se mostrarán en la pantalla del usuario las sustancias que concuerdan con la búsqueda realizada. En esta pantalla de resultados se seguirá mostrando el formulario de búsqueda y la búsqueda anterior realizada, por si se quiere realizar otra búsqueda al visualizar los resultados de la anterior sin tener que volver atrás.

Flujo Alternativo:

2. Si no hay ningún resultado de búsqueda que concuerde con los parámetros introducidos no se mostrará ningún resultado pero se seguirán mostrando los datos por los que se han buscado.

Pos-condición: El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.

11.4.1.17. Añadir Contenido

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere añadir un nuevo contenido al inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Contenidos.

Flujo Básico:

1. El usuario acciona el botón "Add Content".
2. El sistema mostrará al usuario la vista con el formulario con los datos necesarios a introducir.
3. El usuario selecciona las sustancias que desea en este nuevo contenido y añadirá los datos. Posteriormente envía el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el contenido al sistema y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de contenidos.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido un nuevo contenido al sistema.

11.4.1.18. Editar Contenido

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere editar un contenido creado

anteriormente.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de detalle del sustancia frente a un listado de contenidos.

Flujo Básico:

1. El usuario apretará "Edit" en el contenido del listado que quiera editar.
2. El sistema mostrará al usuario la vista con el formulario de edición con los datos del contenido seleccionado.
3. El usuario cambiará los datos que considere oportuno y enviará el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el contenido al sistema con los nuevos datos y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de contenidos.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha editado el contenido seleccionado.

11.4.1.19. Eliminar Contenido

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere eliminar un contenido existente en el sistema

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la

página de edición del contenido.

Flujo Básico:

1. El usuario seleccionara el icono de la papelera.
2. El contenido seleccionado se eliminará del sistema.

Flujo Alternativo:

1. El usuario cancela la operación y vuelve a la vista de sustancias.

Pos-condición: El usuario ha eliminado el contenido seleccionado.

11.4.1.20. Realizar búsqueda interna de Contenidos

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere realizar una búsqueda por las características de un contenido.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Contenidos.

Flujo Básico:

1. El usuario introduce la cadena de texto que desea y pulsa el botón "Search Content" para realizar la búsqueda.
2. Si la búsqueda da resultados, se mostrarán en la pantalla del usuario los contenidos que concuerdan con la búsqueda realizada. En esta pantalla de resultados se seguirá mostrando el formulario de búsqueda y la búsqueda anterior realizada, por si se quiere realizar otra búsqueda al visualizar los resultados de la anterior sin tener que volver atrás.

Flujo Alternativo:

2. Si no hay ningún resultado de búsqueda que concuerde con los parámetros introducidos no se mostrará ningún resultado pero se seguirán mostrando los datos por los que se han buscado.

Pos-condición: El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.

Para simplificar la documentación en los siguientes apartados se incluirán los mismos casos de uso textuales las Localizaciones y Contenedores ya que sus casos de uso son muy parecidos entre ellos.

11.4.1.21. Añadir Localizaciones o Contenedores

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere añadir un nuevo contenedor o localización al inventario.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página de Contenedores o Localizaciones.

Flujo Básico:

1. El usuario acciona el botón "Add Container / Location".
2. El sistema mostrará al usuario la vista con el formulario con los datos necesarios a introducir.
3. El usuario introducirá los datos necesarios y enviará el formulario
4. Si la validación de los datos introducidos es correcta, se añade el contenedor o la localización al sistema y se da un mensaje para avisar al usuario de que la operación

ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de contenedores o localizaciones.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido correctamente un contenedor o una localización al sistema.

11.4.1.22. Editar Localizaciones / Contenedores

Actores: Usuario Técnico y usuario Administrador.

Descripción: Usuario técnico o de administración quiere editar un contenedor o una localización creada anteriormente.

Pre-condición: Haber accedido mediante el login de Técnico o Administrador y encontrarse en la página contenedor o localizaciones frente a un listado de estas.

Flujo Básico:

1. El usuario apretará "Edit" en el contenedor o localización del listado que quiera editar.
2. El sistema mostrará al usuario la vista con el formulario de edición con los datos a editar.
3. El usuario cambiará los datos que considere oportuno y enviará el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el contenedor o localización al sistema con los nuevos datos y se da un mensaje para avisar al usuario

de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista anterior.

4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha editado el contenedor o localización con éxito.

11.4.1.23. Eliminar Localizaciones / Contenedores

Actores: Usuario Administrador.

Descripción: Usuario de administración quiere editar un contenedor o localización creado anteriormente.

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página de edición del elemento.

Flujo Básico:

1. El usuario seleccionara el icono de la papelera.

2. El elemento seleccionado se eliminará del sistema.

Flujo Alternativo:

1. El usuario cancela la operación y vuelve a la vista anterior.

Pos-condición: El usuario ha eliminado el contenedor o localización seleccionado.

Para simplificar la documentación en los siguientes apartados se incluirán los mismos casos de uso textuales los elementos de seguridad Frases de Seguridad, Pictogramas y Características ya que tienen en común los actores que los realizan y sus casos de

uso son muy parecidos entre ellos.

11.4.1.24. Añadir Frases de Seguridad, Pictogramas o Características

Actores: Usuario Administrador.

Descripción: Usuario Administrador que quiere añadir un elemento de seguridad al sistema.

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página del elemento de seguridad correspondiente.

Flujo Básico:

1. El usuario acciona el botón "Add Security".
2. El sistema mostrará al usuario la vista con el formulario con los datos necesarios a introducir.
3. El usuario introducirá los datos necesarios y enviará el formulario
4. Si la validación de los datos introducidos es correcta, se añade el elemento de seguridad al sistema y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista de seguridad.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha añadido correctamente un elemento de seguridad en el sistema.

11.4.1.25. Editar Frases de Seguridad, Pictogramas o Características

Actores: Usuario Administrador.

Descripción: Usuario administrador quiere editar un elemento de seguridad

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página de un elemento de seguridad.

Flujo Básico:

1. El usuario apretará "Edit" en elemento de seguridad del listado que quiera editar.
2. El sistema mostrará al usuario la vista con el formulario de edición con los datos a editar.
3. El usuario cambiará los datos que considere oportuno y enviará el formulario.
4. Si la validación de los datos introducidos es correcta, se añade el elemento al sistema con los nuevos datos y se da un mensaje para avisar al usuario de que la operación ha salido correctamente.

Flujo Alternativo:

3. El usuario cancela la operación y vuelve a la vista anterior.
4. Si la validación de datos es incorrecta, se volverá a la vista anterior volviendo al punto 2 y se mostrará al usuario los errores cometidos.

Pos-condición: El usuario ha editado un elemento de seguridad con éxito.

11.4.1.26. Eliminar Frases de Seguridad, Pictogramas o Características

Actores: Usuario Administrador.

Descripción: Usuario de administración quiere eliminar un elemento de seguridad creado anteriormente.

Pre-condición: Haber accedido mediante el login de Administrador y encontrarse en la página de edición del elemento.

Flujo Básico:

1. El usuario seleccionara el icono de la papelera.
2. El elemento seleccionado se eliminará del sistema.

Flujo Alternativo:

1. El usuario cancela la operación y vuelve a la vista anterior.

Pos-condición: El usuario ha eliminado correctamente el elemento seleccionado.

11.4.1.27. Logout Aplicación

Actores: Usuario Técnico o usuario Administrador.

Descripción: Usuario identificado en la aplicación quiere desconectarse del sistema.

Pre-condición: Estar identificado en el sistema.

Flujo Básico:

1. El usuario presiona el botón logout de la barra superior de opciones.
2. El sistema deslogueará al usuario de la aplicación y le sacará de la sesión en la que se encontraba.

Flujo Alternativo: -

Pos-condición: El usuario identificado pasa a convertirse en un usuario sin identificar.

